IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mohamed Thoufique Ziyavudeen

31/12/2024

Outline

Executive Summary

- **Summary of methodologies**
  - Problem Statement – Could the price of rocket launch be predicted if the launch can successfully land the first stage?
  - Data collection – Data is collected through SpaceX REST API and web scraping Wikipedia website
  - Data Wrangling – Labels (target variables) are created via landing outcome feature
  - Exploratory Data Analysis (EDA) – Data is explored through SQL queries and Data visualization charts
  - Data Visualization – Interactive charts and dashboards are created through Folium and Plotly Dash
  - Model training/prediction – Machine learning models are created to predict landing outcome. Models' hyperparameters are tuned to identify best performing model
- **Summary of all results**
  - KSC LC 39A site is the best site for launching where as CCAFS SLC has roughly 50% success rate
  - Launch sites are located near to coastlines and far away from cities.
  - 1st stage landing success improved over time
  - All machine learning models have the same accuracy at 83% thus it can be used to evaluate the success of 1st stage landing

Introduction

- SpaceX is the first company in the space industry to successfully reuse the first stage.
- SpaceX’s ability to reuse first stage has significantly reduced their expenses down to 62 million US dollars whereas other companies still cost above 165 million US dollars.
- Primary objective of this project is to predict the price of a launch.
- By determining if SpaceX will reuse the first stage using available public knowledge, the price of the launch can be calculated for SpaceX itself or any other competitors such as SpaceY.

- Questions to be answered
  - Which launch site has the highest success rate?
  - Are launch sites closely located to coastline and further away from populated areas?
  - Choose a binary classification machine learning model with high accuracy

Section 1

Methodology

Methodology

Executive Summary

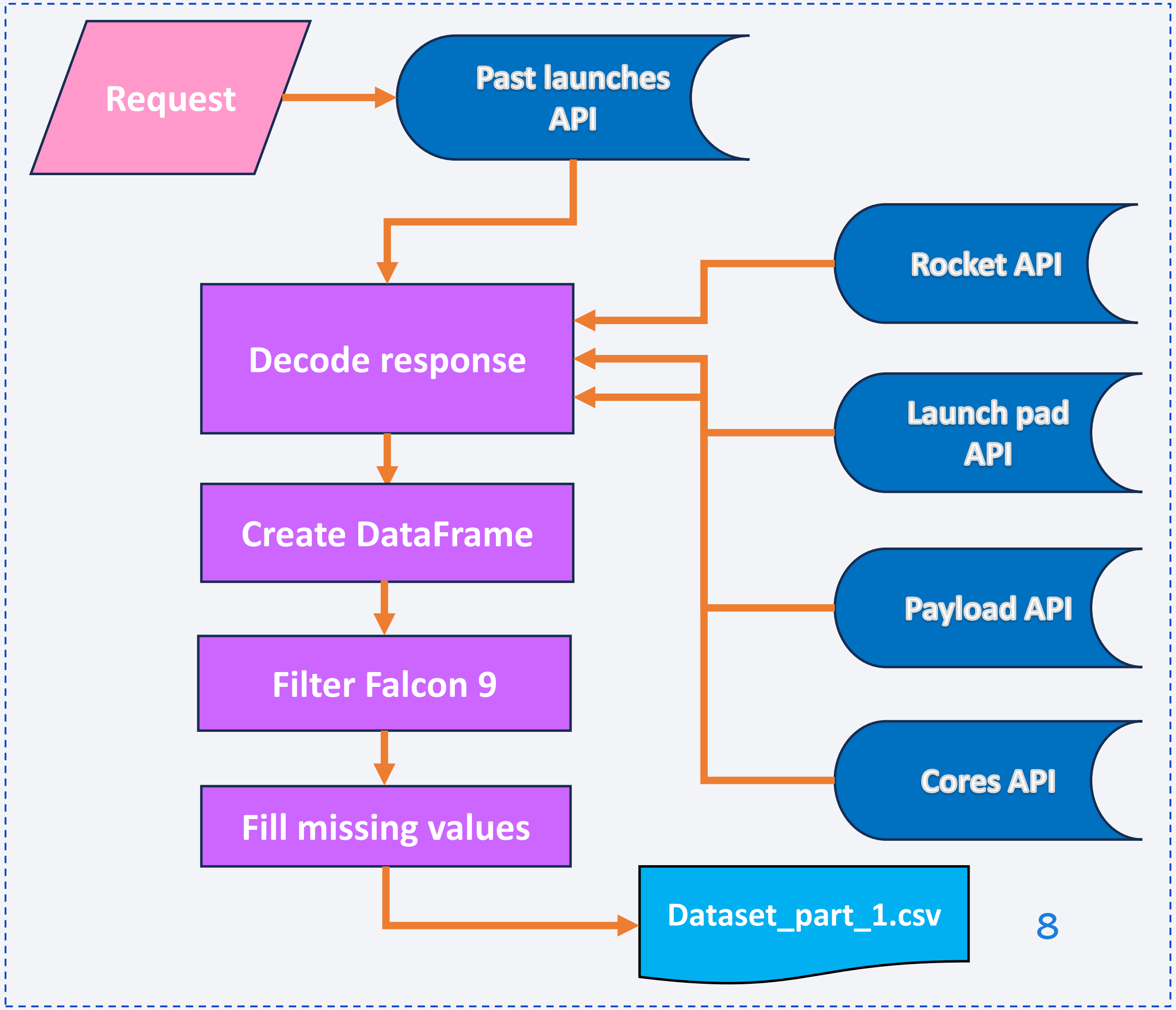
- Data collection:
  - Data collected through API and web scraping.
- Data wrangling
  - Binary variables (0,1) created for successful and unsuccessful 1st stage landings, and missing values are replaced
- Exploratory Data Analysis (EDA) using visualization and SQL
  - Explore data through charts & SQL queries and Feature Engineering is performed
- Interactive visual analytics using Folium and Plotly Dash
  - Map with locations of launch sites, icons to indicate successfulness, and distance from proximities
  - Dashboard to show success rate of launch sites via pie chart, payload size slider and scatter plot to observe patterns
- Predictive analysis using classification models
  - Classification ML models are created, tuned, evaluated using accuracy score and jaccard index

Data Collection

- There were two ways in which the launch data was collected:
  - API
  - Web scraping
- API
  - Initial data was collected through SpaceX's API
  - Several endpoints of API were utilized to get extract relevant data
- Webscraping
  - Data can also be collected through webscraping Wikipedia web page
  - Static url used was https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

Data Collection – SpaceX API - 1

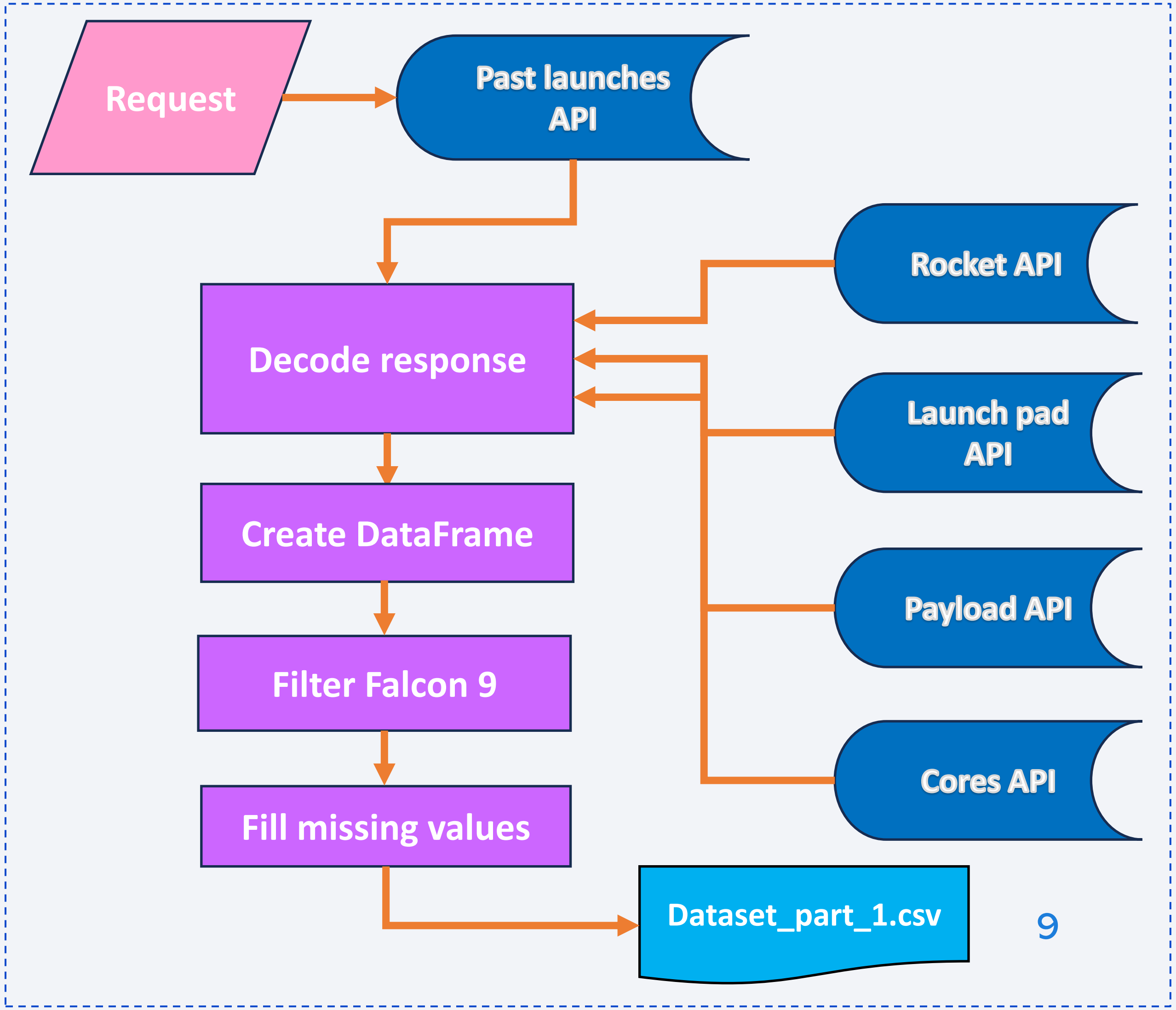
- Raw data was requested from past launches API
- Response returns a list of json objects which then converted into flat table using .json_normalize() method
- Many features had IDs so different endpoints of API were used to mine applicable data
  - Booster version was extracted from Rocket API endpoint
  - Latitude, Longitude and Launch Site were extracted from Launchpads API endpoint
  - Payload Mass (in kgs) and Orbit were extracted from Payload API endpoint
  - Block, Reuse count, Serial, Outcome, Flights, Gridfins, Reused, Legs and Landing pad were extracted from cores API endpoint.



Data Collection – SpaceX API - 2

- Create launch_data DataFrame
- Sampling data via filtering to only view Falcon 9 launches
- Null values of Payload Mass column were replaced by mean of Payload Mass
- Data exported into csv file

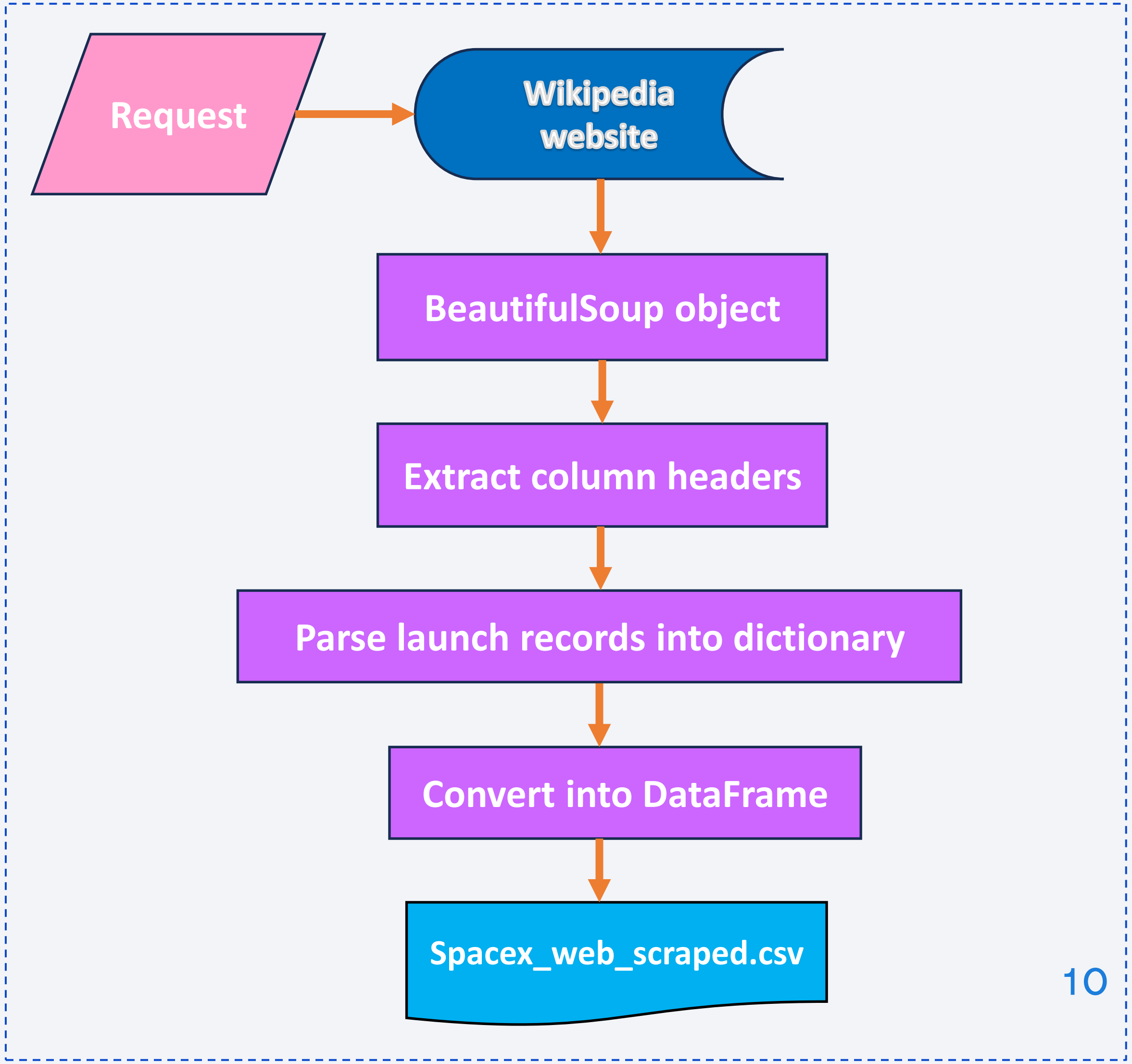
Data_collection.ipynb



Data Collection - Scraping

- Request Falcon 9 launch data from a static url of Wikipedia's webpage
- Create BeautifulSoup object and pass in html content as argument
- Identify the table with launch records
- Extract column headers from the launch table
- Create a dictionary and parse launch records into the dictionary
- Convert dictionary into pandas DataFrame
- Export DataFrame into csv file

[Webscraping.ipynb]



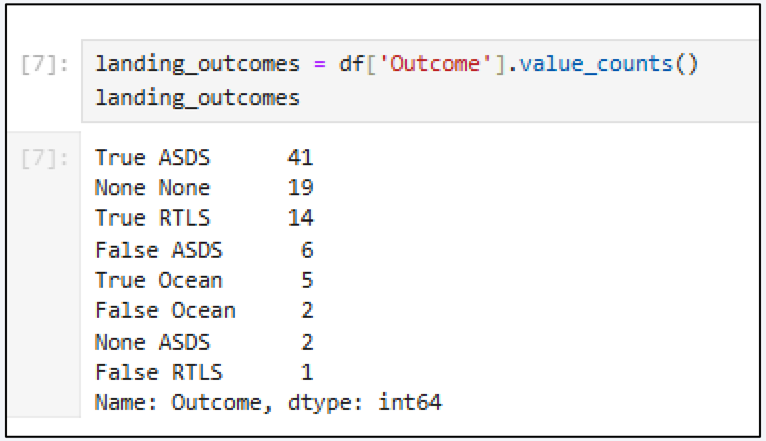
Data Wrangling - 1

Perform EDA

- Calculate unique launch sites
  - **CCAFS SLC 40** (Cape Canaveral Space Launch Complex 40)
  - **KSC LC 39A** (Kennedy Space Center Launch Complex 39A)
  - **VAFB SLC 4E** (Vandenberg Air Force Base Space Launch Complex 4E)
- List different types of orbits
- Identify unique landing outcomes

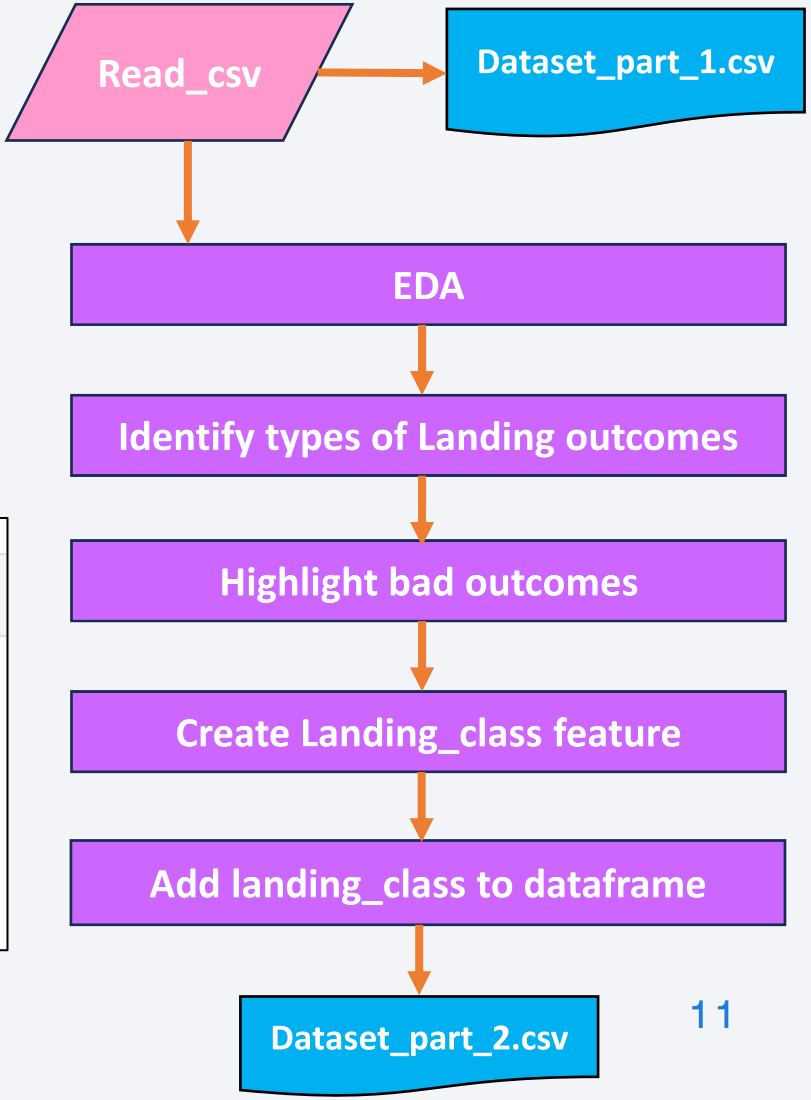
There are three types of location where 1st stage could be landed

- **Ocean** – Land on a specific area of the ocean
- **RTLS** – Land on a Ground pad
- **ASDS** – Land on a Drone ship



```
[7]: landing_outcomes = df['Outcome'].value_counts()
     landing_outcomes
```

```
[7]: True ASDS      41
     None None      19
     True RTLS      14
     False ASDS      6
     True Ocean      5
     False Ocean     2
     None ASDS       2
     False RTLS      1
     Name: Outcome, dtype: int64
```

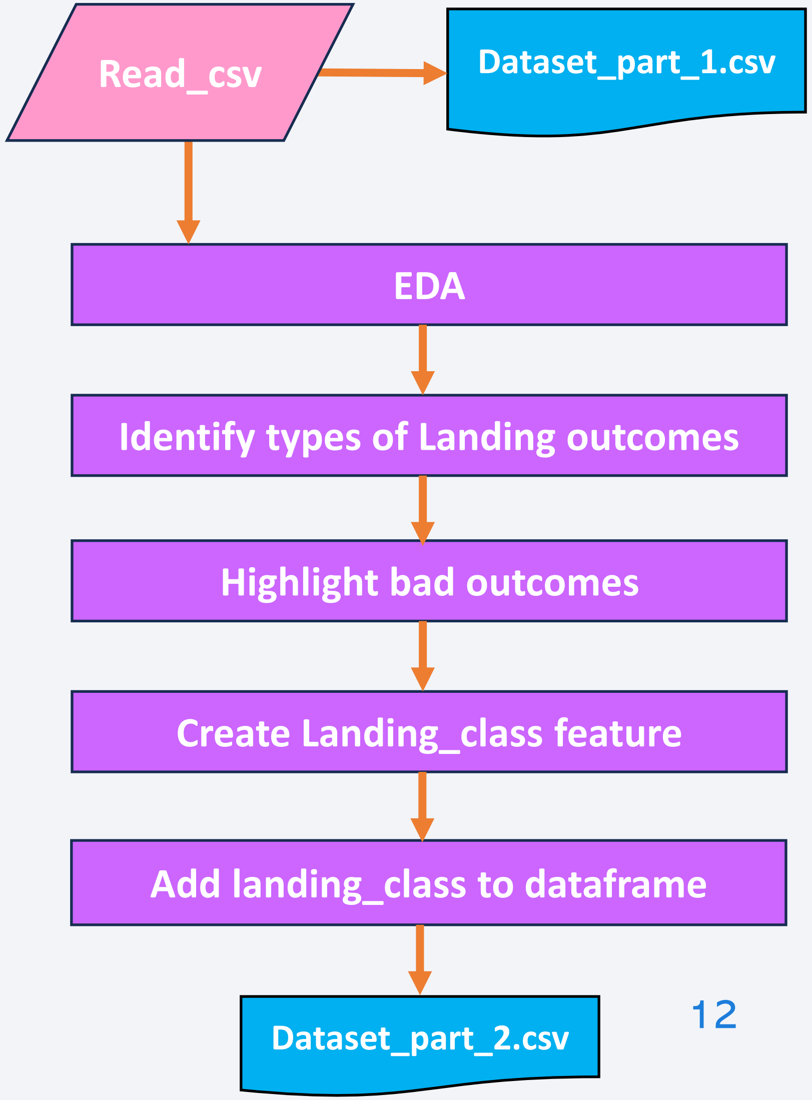


Read_csv
Dataset_part_1.csv
EDA
Identify types of Landing outcomes
Highlight bad outcomes
Create Landing_class feature
Add landing_class to dataframe
Dataset_part_2.csv

Data_wrangling.ipynb

Data Wrangling - 2

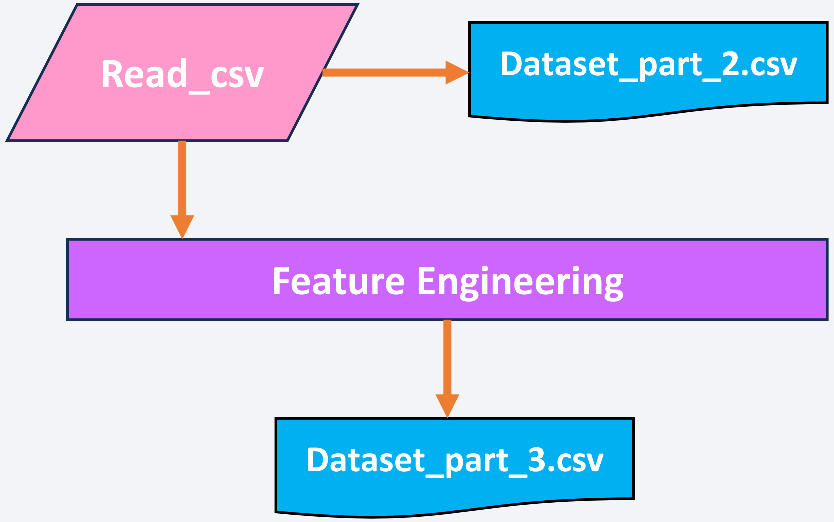
- Define labels for supervised learning models
- Successful outcome (1st stage landed successfully)
  - True ASDS
  - True RTLS
  - True Ocean
- Unsuccessful outcome (1st stage did NOT land)
  - False ASDS
  - False RTLS
  - False Ocean
  - None None
- Using the Outcome feature, landing class feature is created
  - If the outcome is successful, labelled as 1
  - If the outcome is unsuccessful, labelled as 0
- Landing_class feature is then added to dataframe


Read_csv
Dataset_part_1.csv
EDA
Identify types of Landing outcomes
Highlight bad outcomes
Create Landing_class feature
Add landing_class to dataframe
Dataset_part_2.csv

Data_wrangling.ipynb

EDA with Data Visualization

1) Number of flights vs Launch site – A scatter plot to check the performance of launch site over time with points colour coded as successful and unsuccessful 1st stage landings
2) Payload vs Launch site - A scatter plot to identify which sites are suitable for smaller, medium and heavy sized payloads
3) Success rate of Orbit types - A bar chart to compare which orbit type is more successful
4) Number of flights vs Orbit type - A scatter plot to capture insight on whether orbit type has an impact on the class
5) Payload vs Orbit type - A scatter plot to check if there is a relationship between payload and orbit type. Can high altitude orbits sustain heavier payloads?
6) Launch success yearly trend - A line chart to find out if there is a trend in the average success rate every year



Feature Engineering is performed in EDA to convert categorical features into numerical features since Machine Learning models require numerical data.

EDA_Data_Visualization.ipynb

EDA with SQL

1. Unique launch sites
2. 5 records of launch sites of Cape Canaveral Space Launch Complex (begin with ‘CCA’)
3. Calculate total payload mass supported by boosters launched by NASA (CRS)
4. Calculate average payload mass carried by booster version F9 v1.1
5. When was the 1st successful landing outcome in ground pad was achieved?
6. For a payload mass between 4000 and 6000 kg, find the successful drone ship boosters.
7. List the total number of successful and failed mission outcomes
8. List booster versions that have carried maximum payload mass.
9. Date, Booster version and Launch site of failed (drone ship) outcomes in 2015
10. Summarize count of landing outcomes between 2010-06-04 and 2017-03-20

EDA_SQL.ipynb

Build an Interactive Map with Folium

- Create a **map** object
- Add **circle** object for each launch site along with **popup** labels to the map object
- For each record, add **marker** object with it's coordinates for identification
- Marker icon property is customized to indicate success or failed launch – Green is 1, and Red is 0
- Markers are concentrated around 4 sites only, so **marker cluster** object is created.
- Markers are added to marker cluster for each record
- Marker cluster is then added to map object
- Add mouse position to identify coordinates of proximities such as coastline line, railway, city and highway.
- Calculate the distance between launch site and proximities
- Add polyline (straight line) between launch site and proximities along with distance marker

Launch_site_location.ipynb

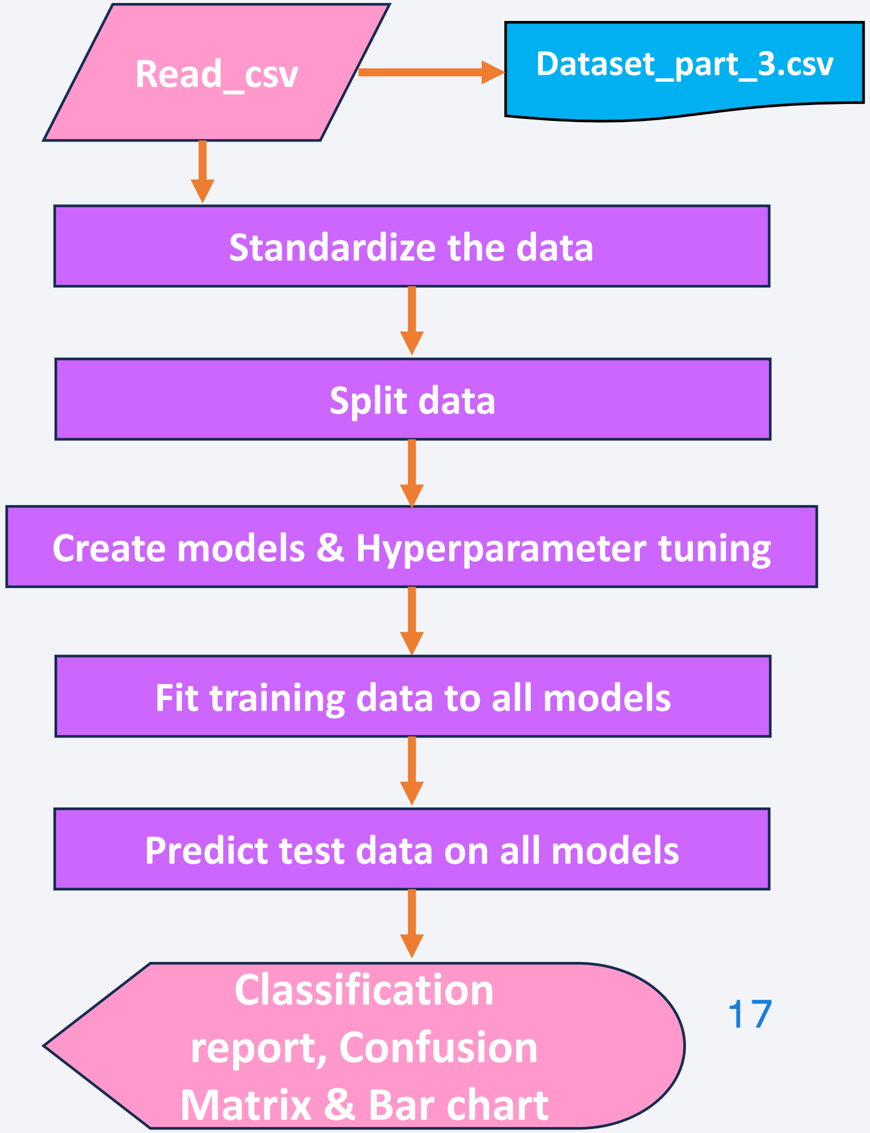
Build a Dashboard with Plotly Dash

- An interactive pie chart to display proportion of successful launches for all sites.
- A drop-down box to select individual sites and the pie chart updates for the chosen site, courtesy of callback function in dash.
- Payload range slider to identify insights such as which payload range is more successful
- Scatter plot to provide comprehension on the relationship between Payload vs Class for the chosen payload range

Spacex_launch_dashboard.py

Predictive Analysis (Classification)

- Independent variables X are features such as flight number, payload, orbit, booster version, etc whereas the dependent (target) variable Y is class (0 - unsuccessful landing, 1 – successful landing)
- Data is standardized then split into training, validation and test data (20%)
- Using GridSearchCV, hyperparameters are tuned for a cross-validation value of 10 for the following machine learning models
  - Logistic regression
  - Support Vector Machine (SVM)
  - Decision Tree
  - K-Nearest Neighbors (KNN)
- Training data is fitted to these models which then predicts class on test data
- Accuracy score and Jaccard index are calculated
- Confusion matrix is plotted for all the models


Read_csv
Dataset_part_3.csv
Standardize the data
Split data
Create models & Hyperparameter tuning
Fit training data to all models
Predict test data on all models
Classification report, Confusion Matrix & Bar chart

[Machine_Learning_models.ipynb]

Results

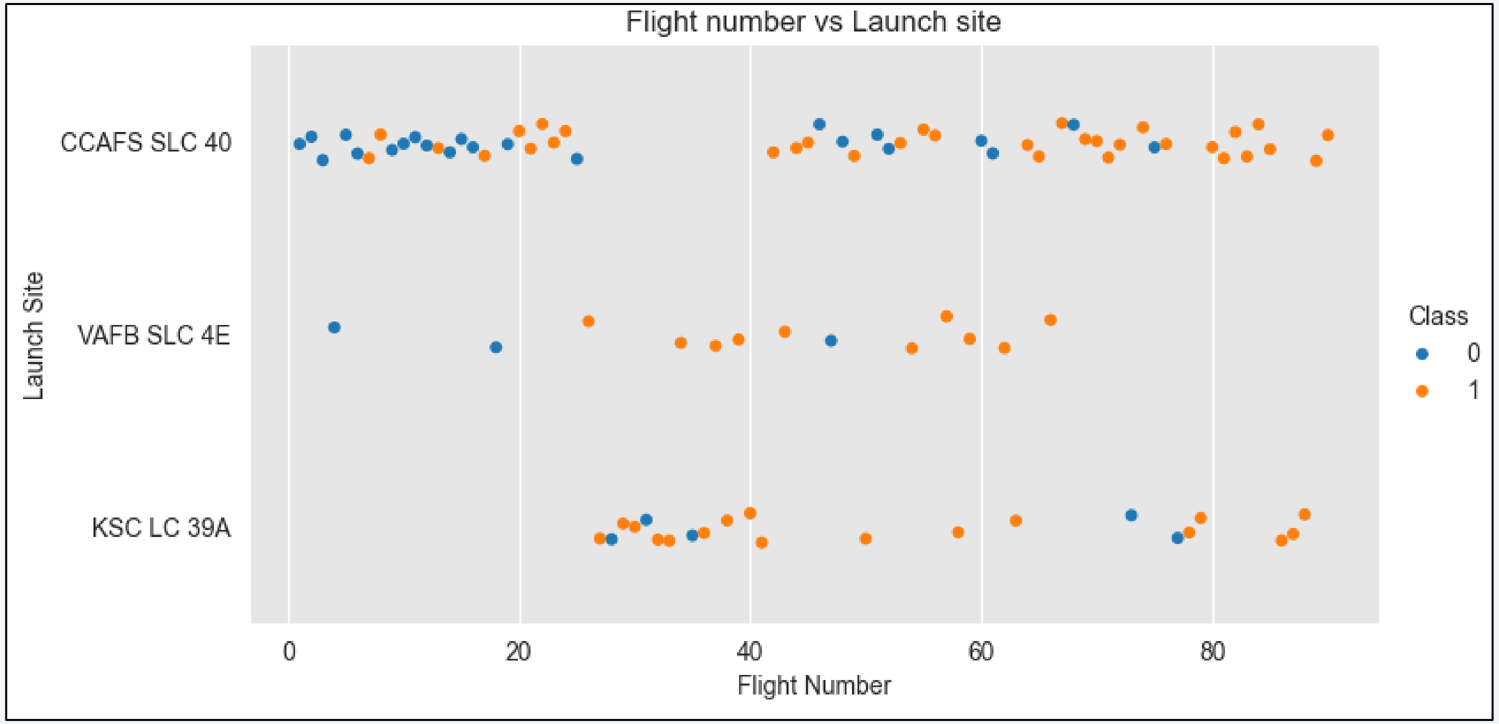
- Exploratory data analysis results
  - Positively upward trend in success rate of 1st stage landing over the years
  - KSC has higher success rate 76.9% compared to CCAFS (30.3%) and VAFB SLC (40%) sites.
- Interactive analytics results
  - Coastline is only less than 1 km away from launch site CCAFS
  - CCAFS Launch site is more than 20 km away from densely populated areas such as cities, as well as transportation routes such as highways and railway tracks.
  - Heavier payloads (>7000 kg) tend to be more successful
  - F9 booster launches are highly successful
- Predictive analysis results
  - All models have 83% accuracy and Jaccard index of 0.8
  - Confusion matrix is the same for all models
  - Average f1 score is 78% for both successful and unsuccessful landings

Section 2

Insights drawn from EDA

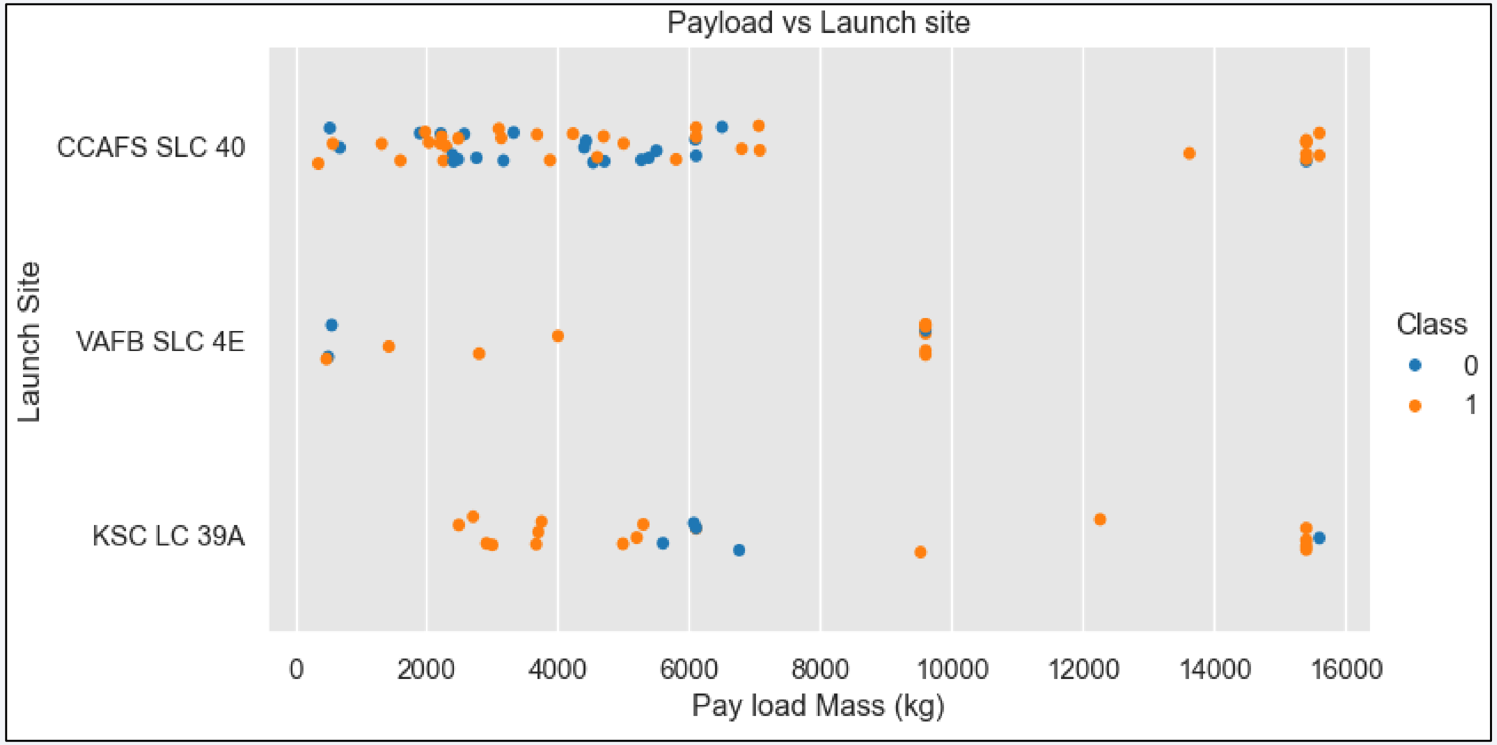
Flight Number vs. Launch Site

- Initial launches had large unsuccessful landings specifically at CCAFS SLC 40 site
- After 20 launches, successful landings were more prominent
- 80+ launches are highly successful
- VAFB SLC and KSC LC sites have higher success rate compared to CCAFS SLC site



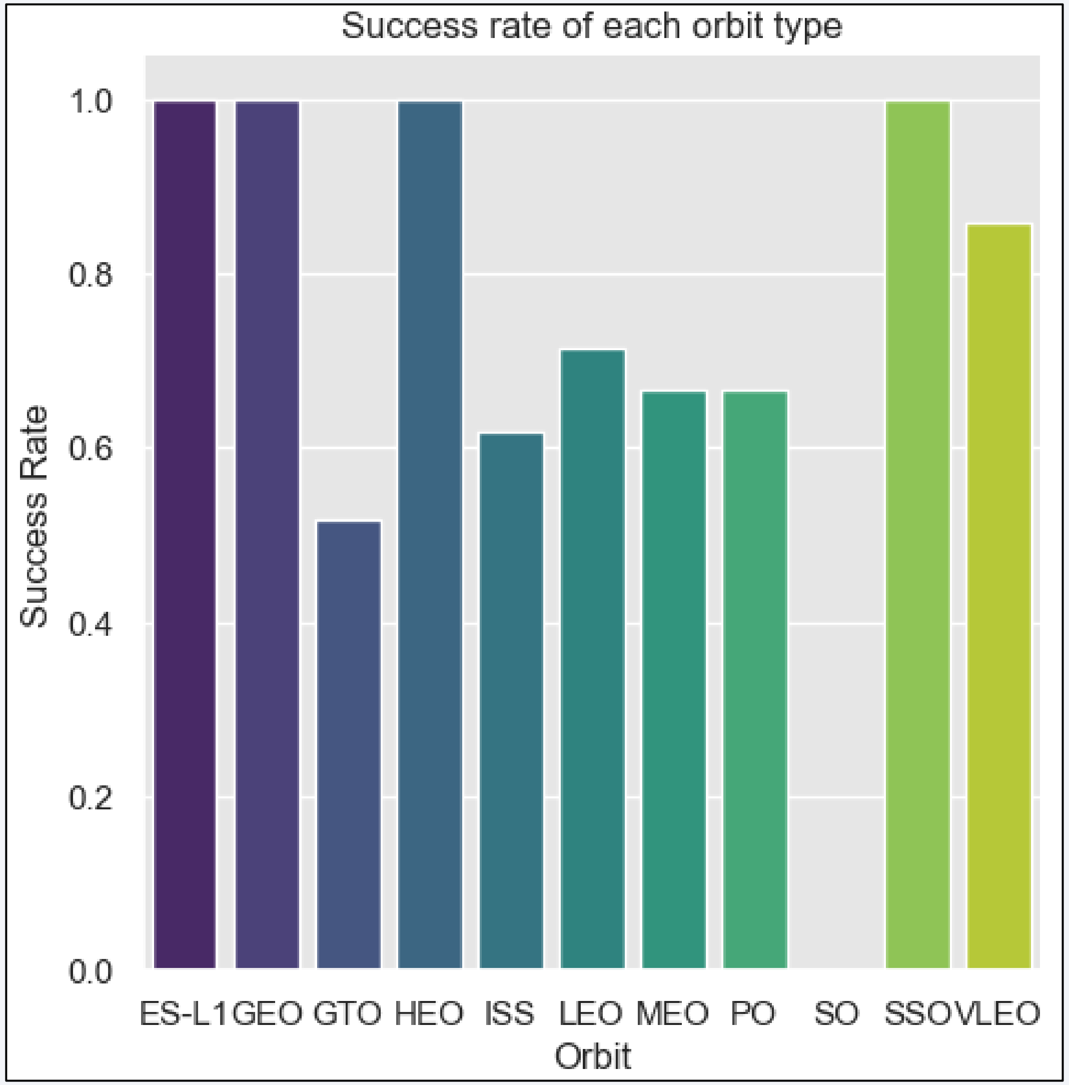
Payload vs. Launch Site

- CCAFS SLC site exhibits mixed proportion of successful and unsuccessful landings for payload less than 8000 kg.
- VAFB SLC site has relatively more successful landings but has not launched payloads higher than 10000 kg
- KSC LC has more success for payload less than 5000 kg and generally more successful for heavier payloads.
- For all sites, payloads more than 7000 kg, launches tend to be more successful



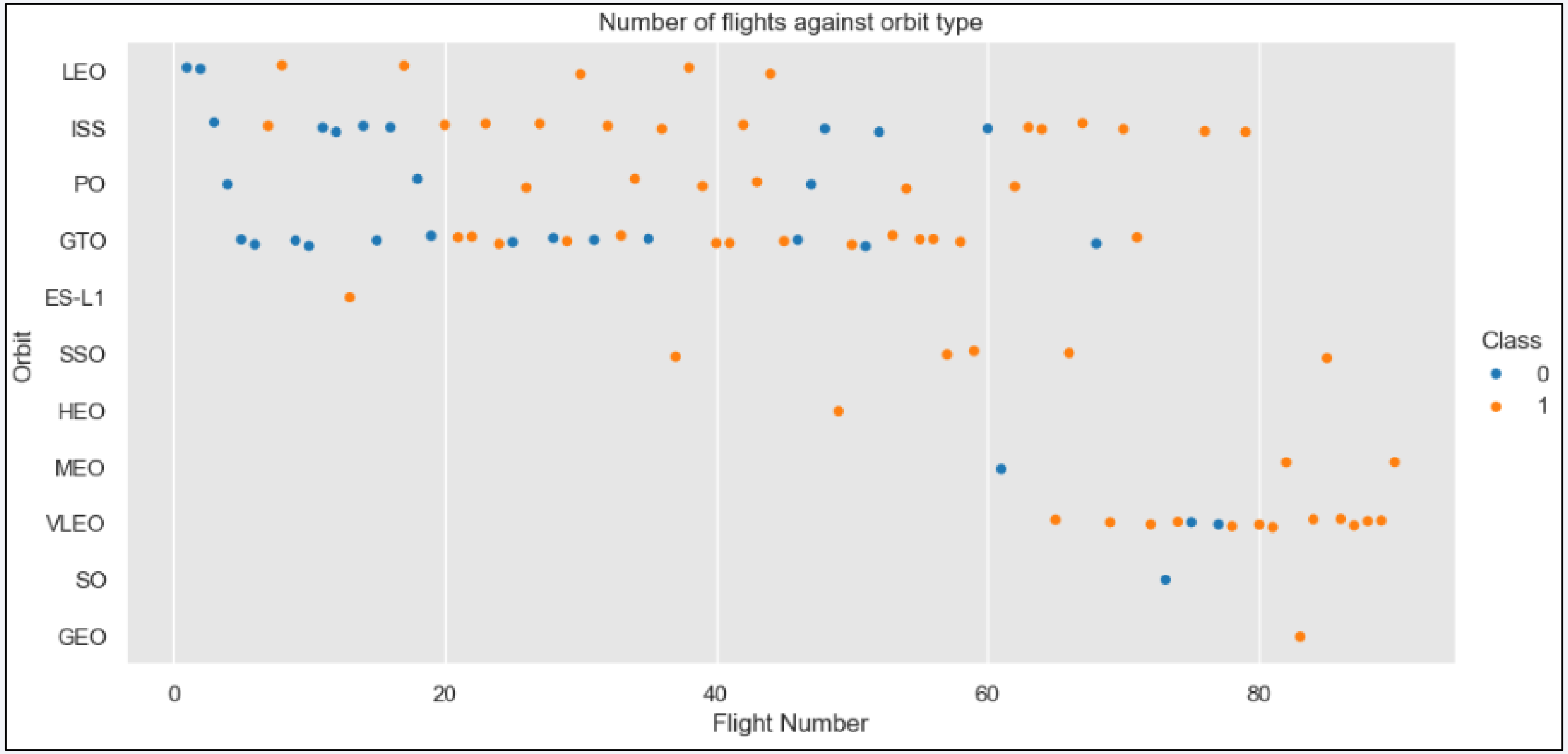
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, and SSO orbits have 100% success rate
- VLEO has 85% success rate
- Other orbits such as GTO, ISS, LEO, MEO and PO range between 50 to 70% success rate
- SO orbit has 0% success rate



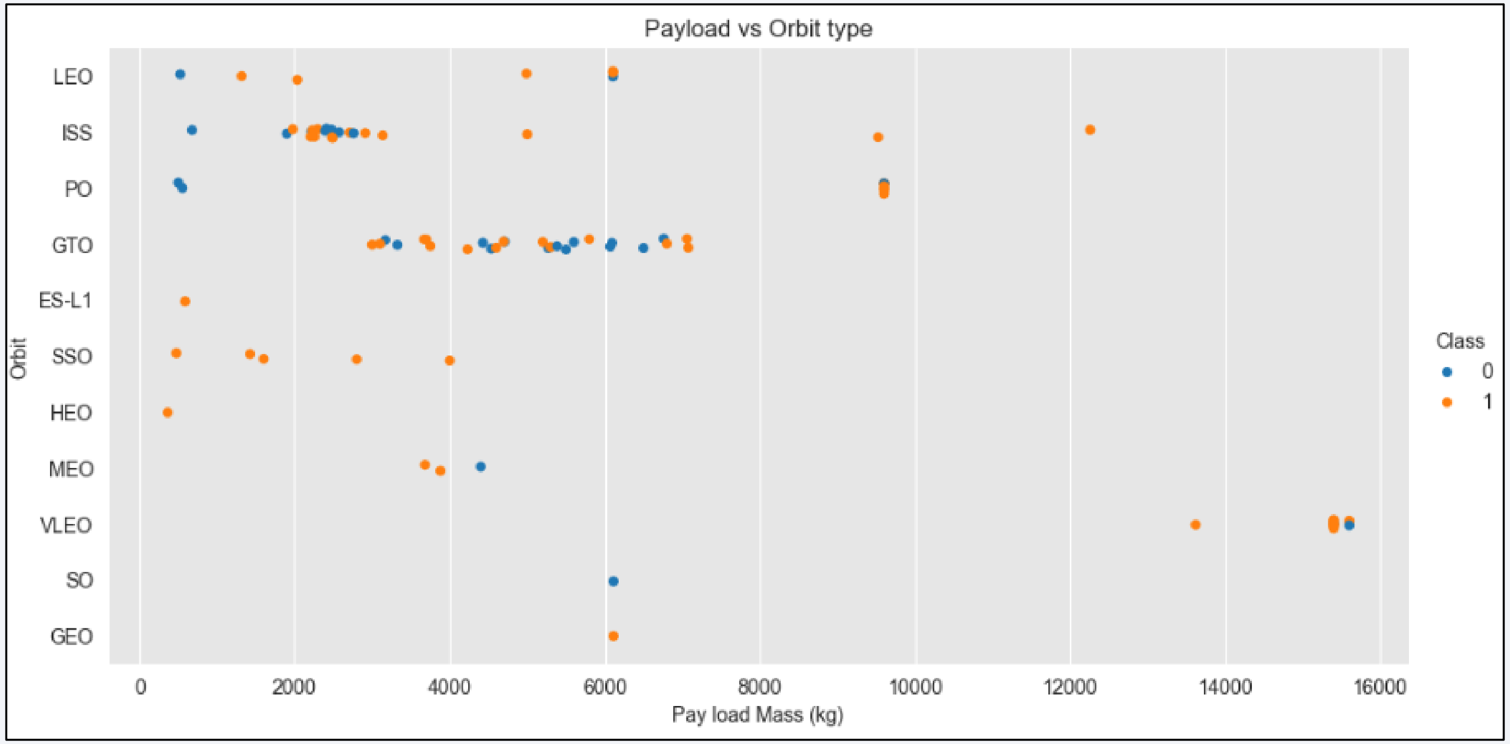
Flight Number vs. Orbit Type

- GTO orbit was the most frequently observed orbit with 51.9% success rate of landing 1st stage
- LEO, GTO, PO and ISS were the most preferred orbits for launches below 60
- VLEO, SSO and LEO have highly successful launches
- After 60 launches, VLEO and ISS were the most preferred orbits with high success



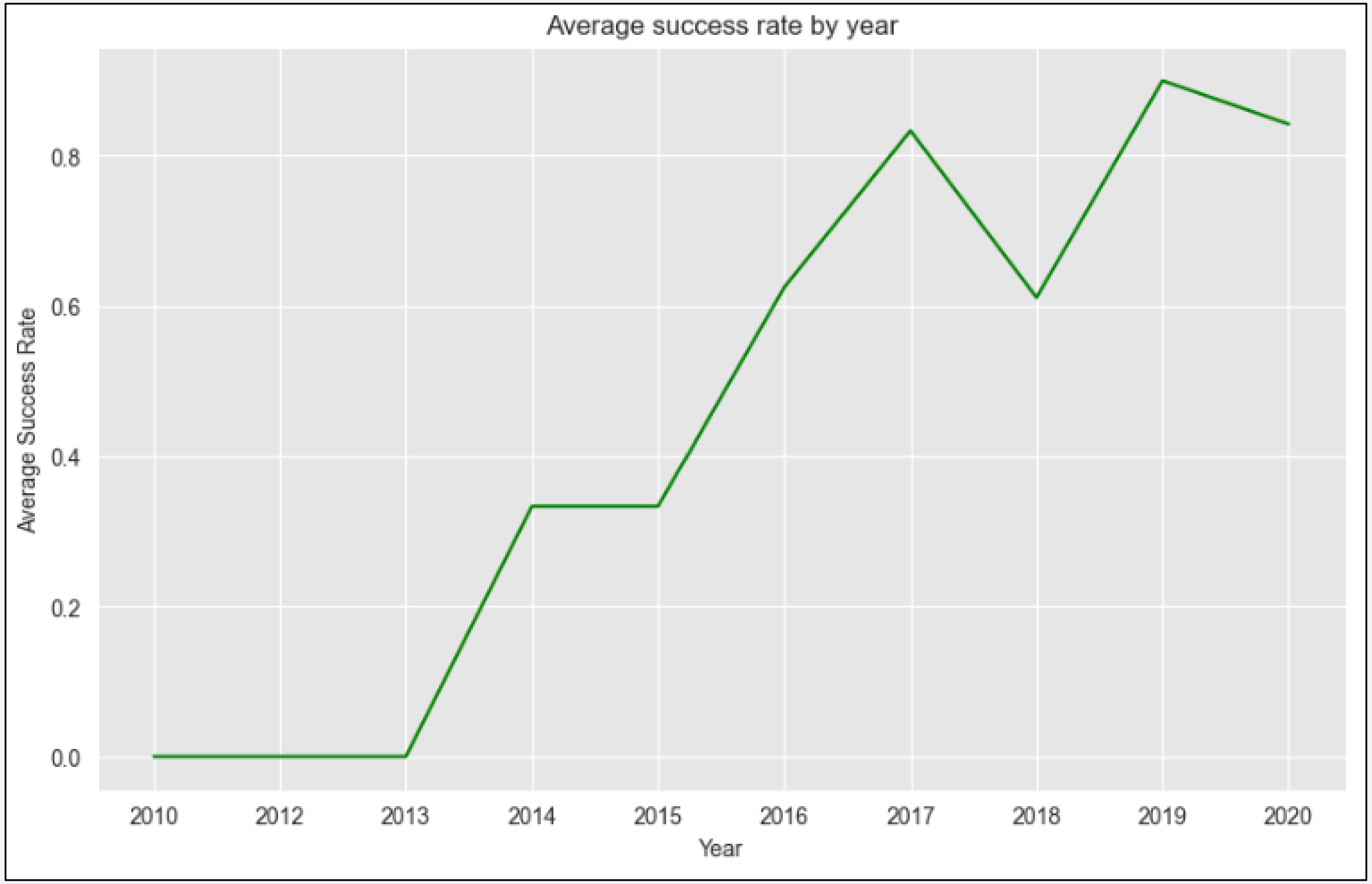
Payload vs. Orbit Type

- SSO has 100% success rate for lower payload range (< 4000 kgs)
- VLEO, PO and ISS are suitable for heavier payloads (>8000 kgs)
- For GTO, it is difficult to differentiate successful and unsuccessful landings.



Launch Success Yearly Trend

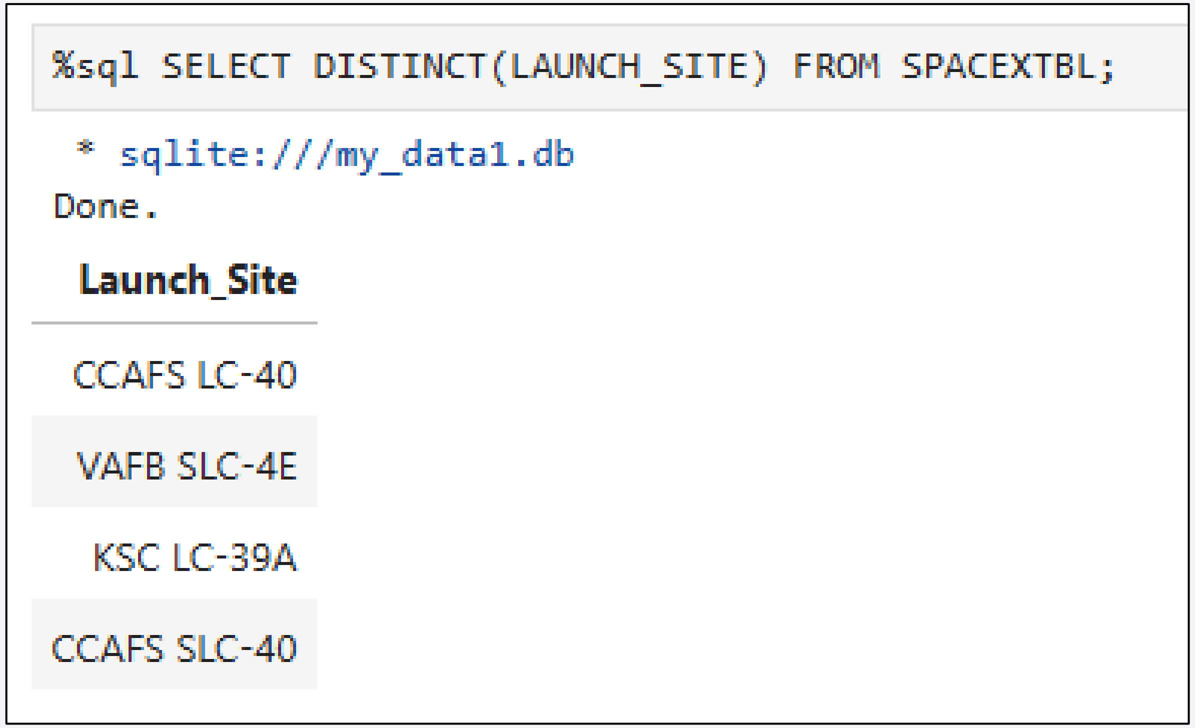
- From 2013, success rate has increased significantly from 0 to 35%.
- A steady climb between 2015 and 2017 peaking just above 80%.
- Fluctuations begin from 2017 but positively going beyond 90%



All Launch Site Names

Unique launch sites are:

- **CCAFS LC-40**
- **VAFB SLC-4E**
- **KSC LC-39A**
- **CCAFS SLC-40**



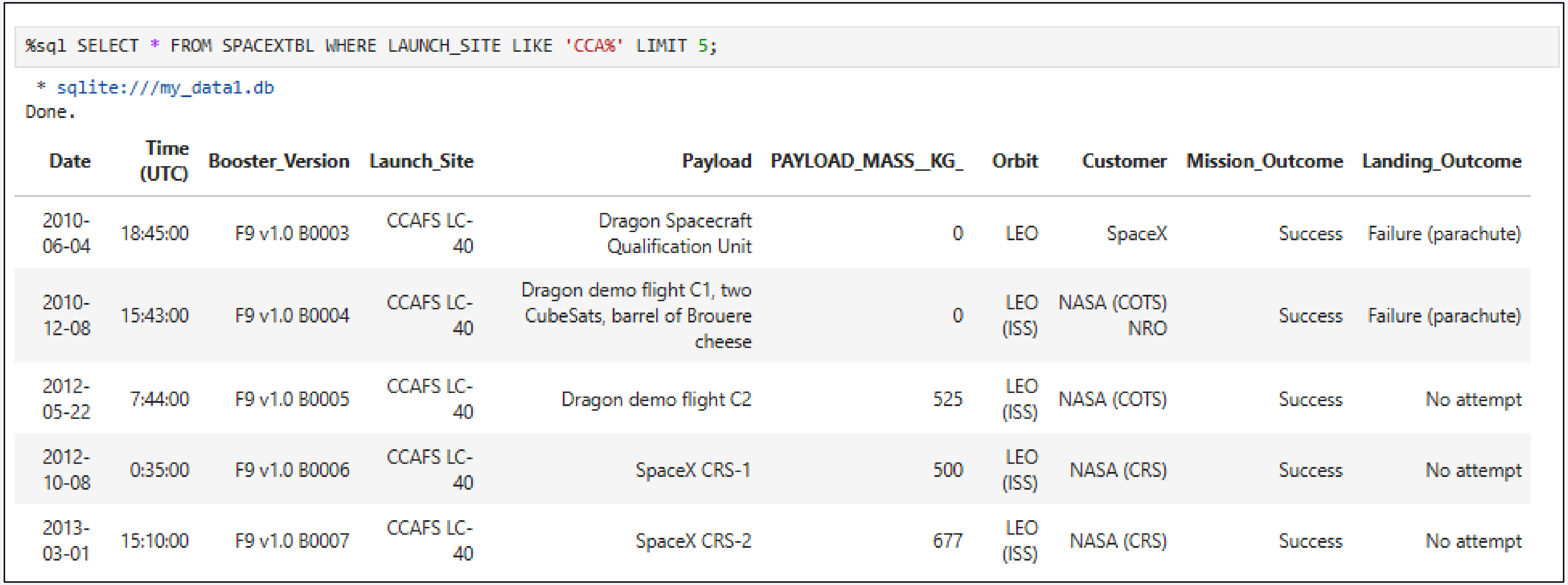
```
%sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL;
```

```
 * sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Site Names Begin with 'CCA'

5 records where launch sites begin with `CCA`



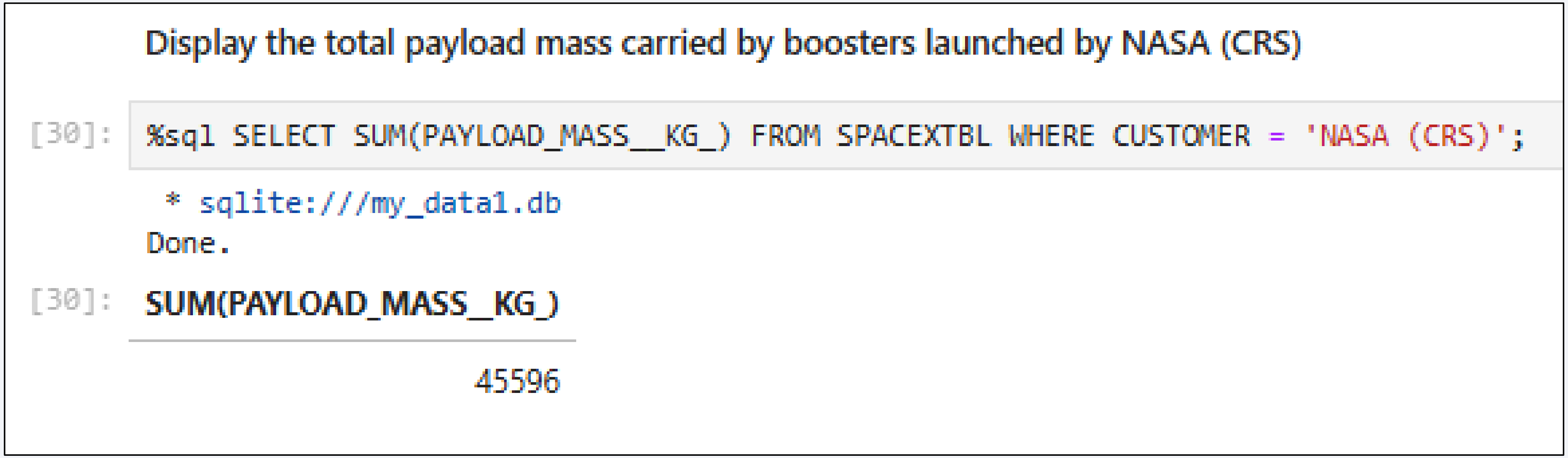
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

Total payload carried by boosters from NASA is **45596 kilograms**



Display the total payload mass carried by boosters launched by NASA (CRS)

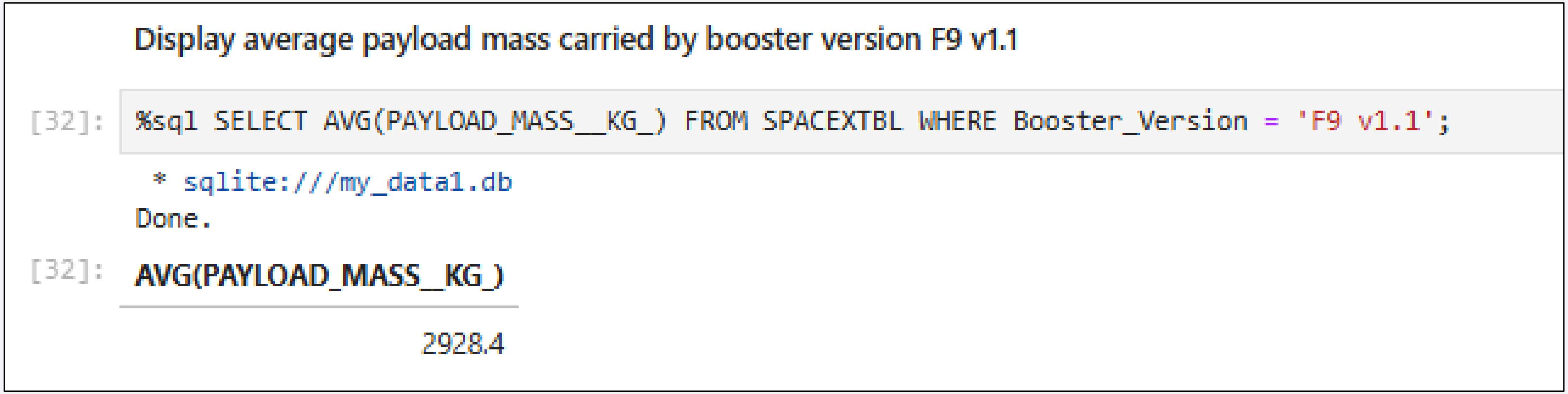
```
[30]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';
```

```
 * sqlite:///my_data1.db
Done.
```

```
[30]: SUM(PAYLOAD_MASS__KG_)
                      45596
```

Average Payload Mass by F9 v1.1

Average payload mass carried by booster version F9 v1.1 is **2928.4 kg**



Display average payload mass carried by booster version F9 v1.1

```
[32]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1';
```

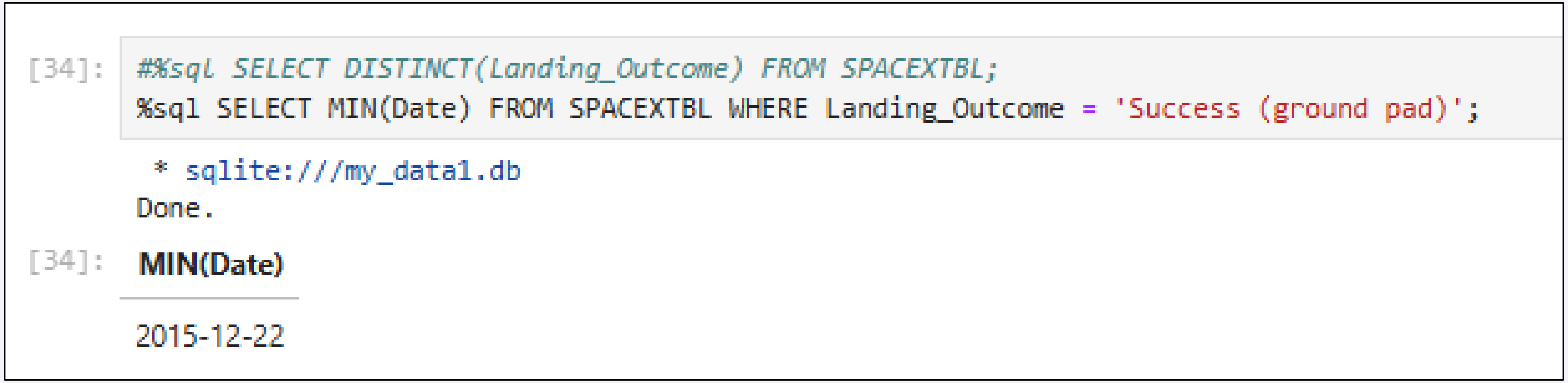
```
 * sqlite:///my_data1.db
Done.
```

[32]:

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

First successful landing outcome on ground pad was on **2015-12-22**



```
[34]: #%sql SELECT DISTINCT(Landing_Outcome) FROM SPACEXTBL;
      %sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)';

       * sqlite:///my_data1.db
      Done.
```

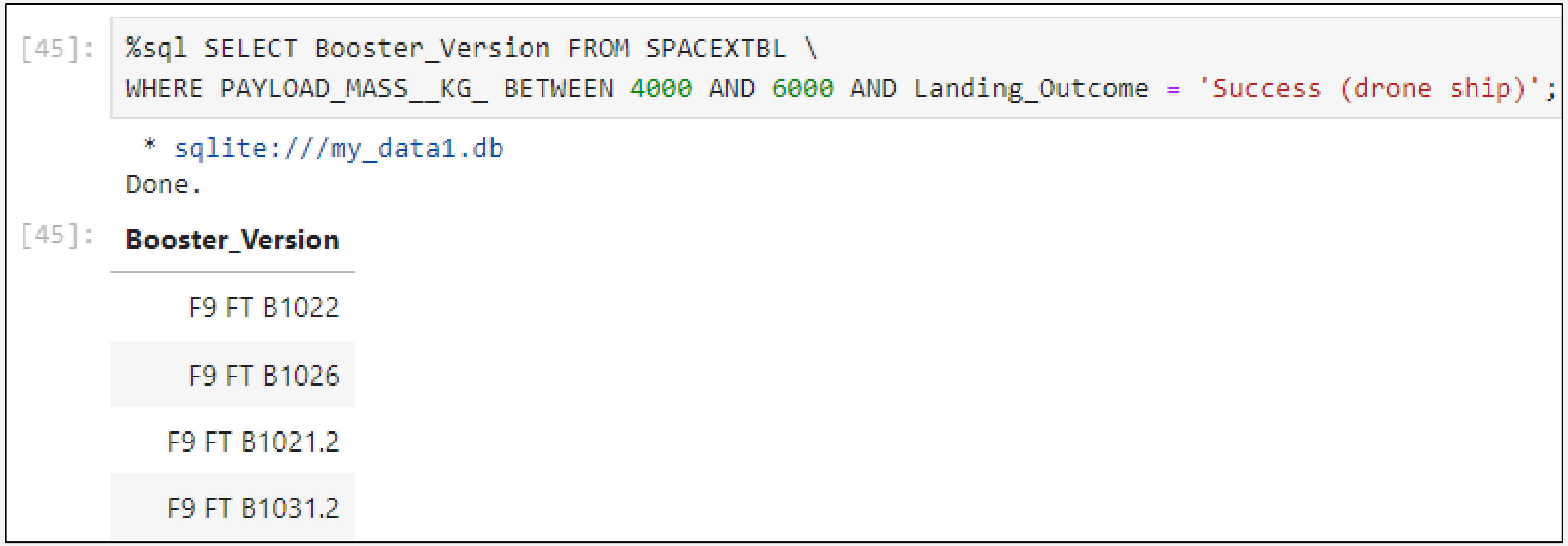
[34]:

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Names of boosters which have successfully landed on drone ship

- **F9 FT B1022**
- **F9 FT B1026**
- **F9 FT B1021.2**
- **F9 FT B1031.2**



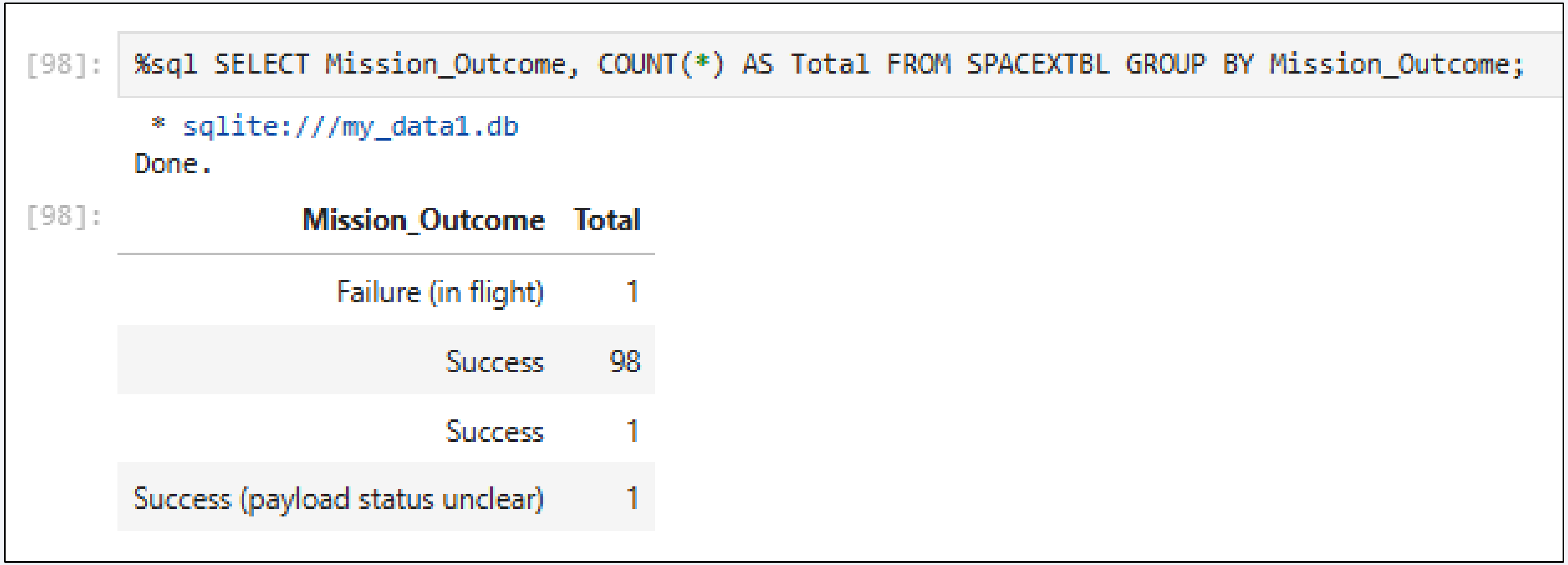
```
[45]: %sql SELECT Booster_Version FROM SPACEXTBL \
      WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND Landing_Outcome = 'Success (drone ship)';

       * sqlite:///my_data1.db
      Done.
```

[45]:

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

Query shows the total number of successful and failure mission outcomes



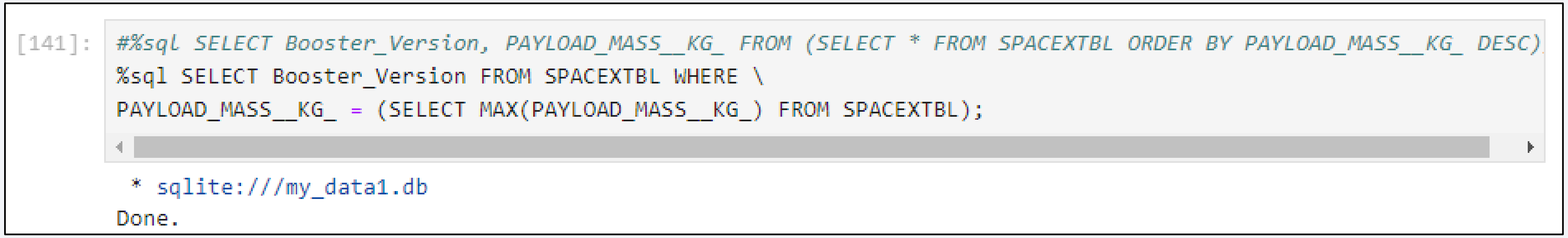
```
[98]: %sql SELECT Mission_Outcome, COUNT(*) AS Total FROM SPACEXTBL GROUP BY Mission_Outcome;

 * sqlite:///my_data1.db
Done.
```

[98]:

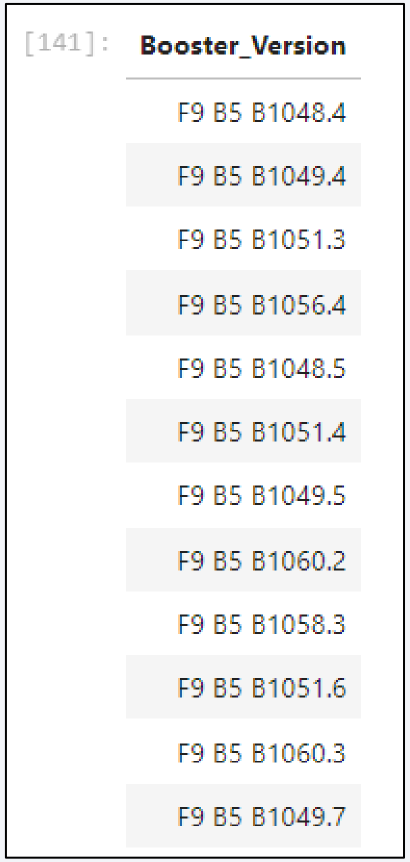
Mission_Outcome	Total
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

Query displays the list of boosters that have carried maximum payload mass via a subquery



```
[141]: #%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM (SELECT * FROM SPACEXTBL ORDER BY PAYLOAD_MASS__KG_ DESC)
       %sql SELECT Booster_Version FROM SPACEXTBL WHERE \
       PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);

 * sqlite:///my_data1.db
Done.
```

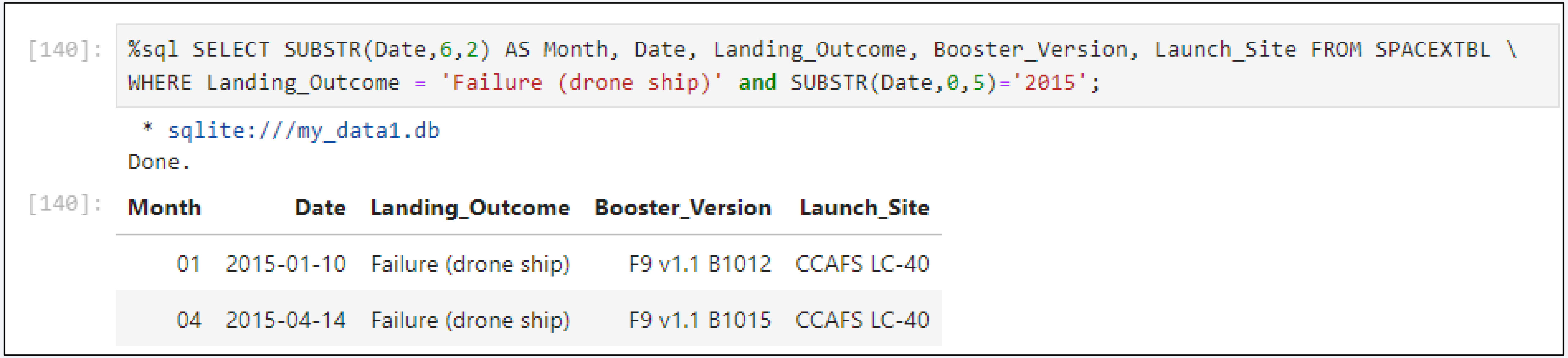


[141]:

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

Query returns failed outcomes in the year 2015 of drone ship, their respective booster versions, and launch site



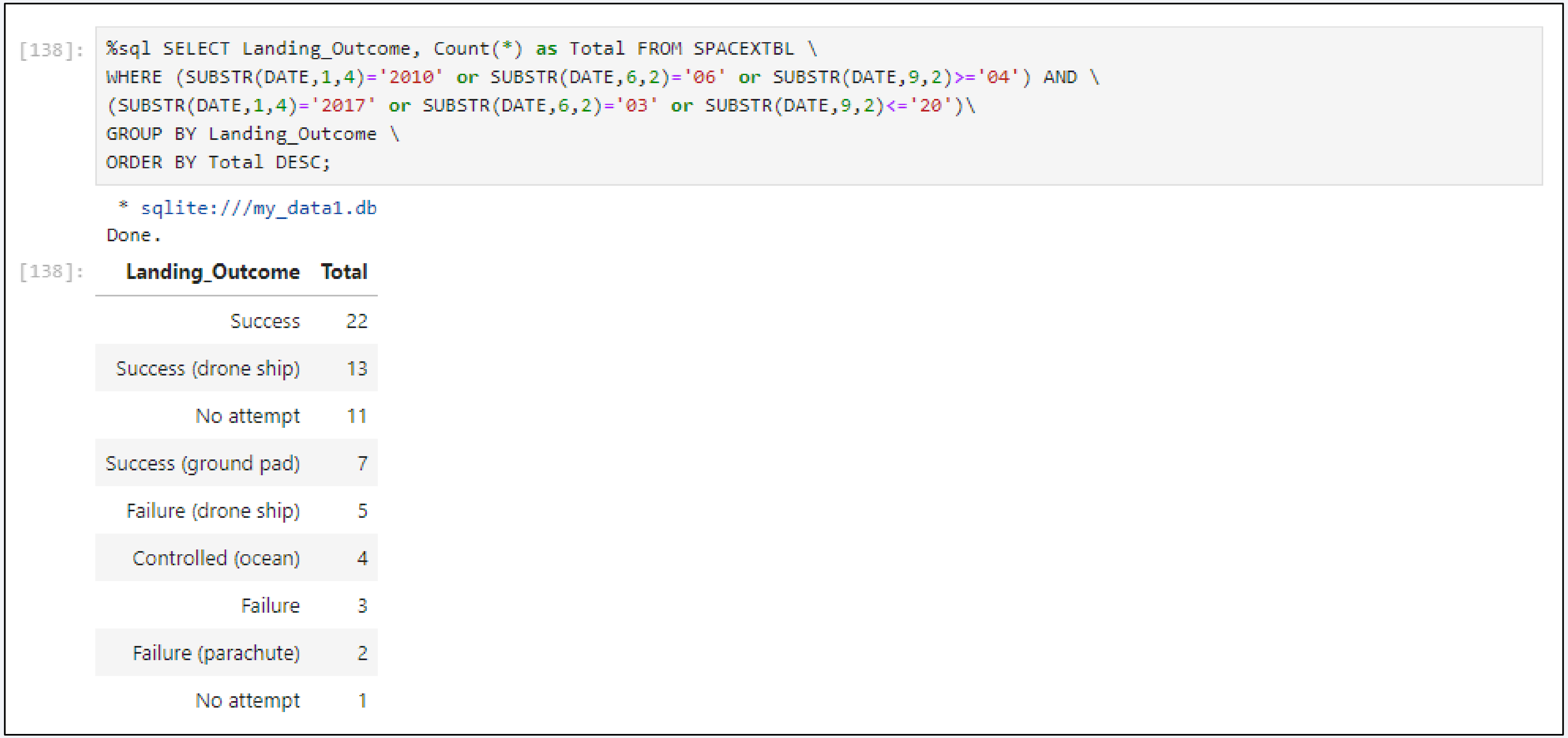
```
[140]: %sql SELECT SUBSTR(Date,6,2) AS Month, Date, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTBL \
       WHERE Landing_Outcome = 'Failure (drone ship)' and SUBSTR(Date,0,5)='2015';

        * sqlite:///my_data1.db
       Done.
```

[140]:

Month	Date	Landing_Outcome	Booster_Version	Launch_Site
01	2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

Query counts landing outcomes between the date 2010-06-04 and 2017-03-20, and displays them in descending order



[138]:

```
%sql SELECT Landing_Outcome, Count(*) as Total FROM SPACEXTBL \
WHERE (SUBSTR(DATE,1,4)='2010' or SUBSTR(DATE,6,2)='06' or SUBSTR(DATE,9,2)>='04') AND \
(SUBSTR(DATE,1,4)='2017' or SUBSTR(DATE,6,2)='03' or SUBSTR(DATE,9,2)<='20')\
GROUP BY Landing_Outcome \
ORDER BY Total DESC;
```

```
 * sqlite:///my_data1.db
Done.
```

[138]:

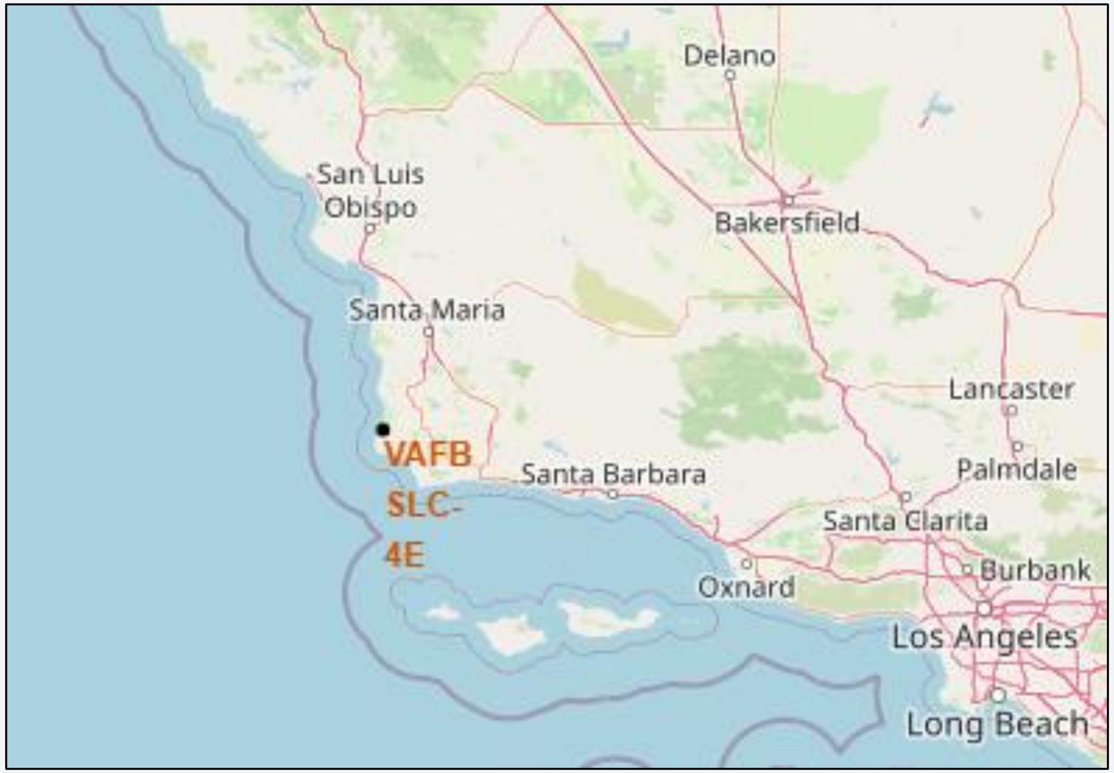
Landing_Outcome	Total
Success	22
Success (drone ship)	13
No attempt	11
Success (ground pad)	7
Failure (drone ship)	5
Controlled (ocean)	4
Failure	3
Failure (parachute)	2
No attempt	1

Section 3

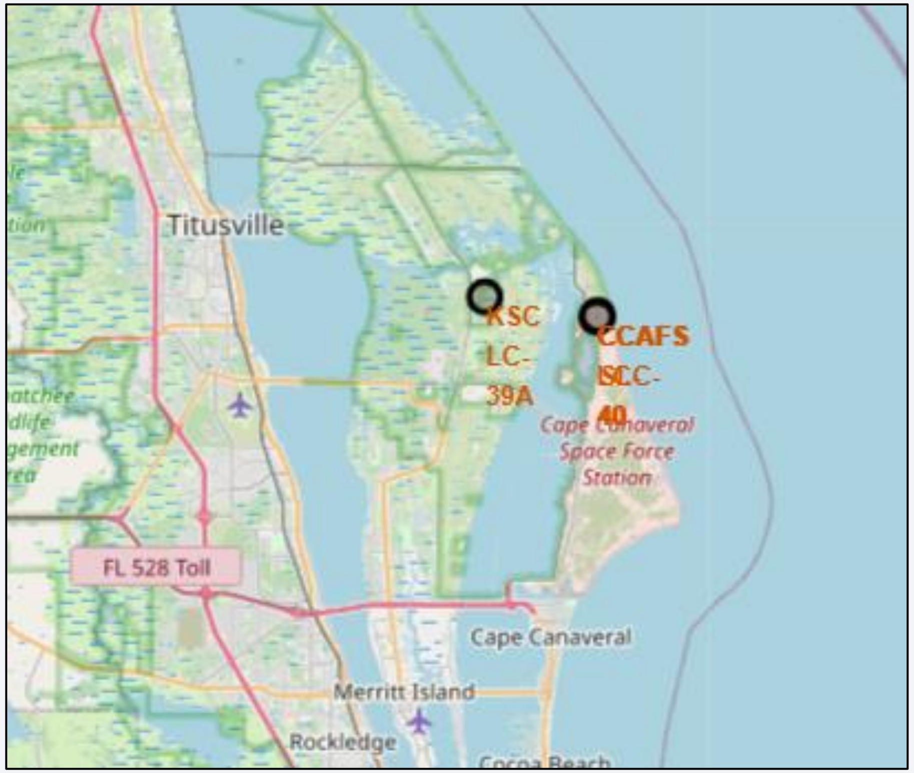
Launch Sites Proximities Analysis

Folium Map – Launch sites markers

Launch sites' location **markers** on a global map

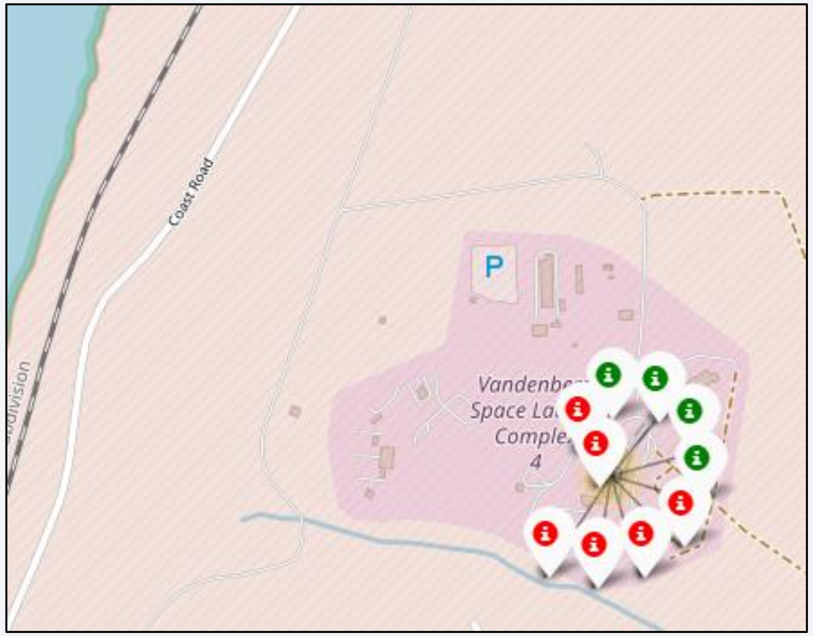


Vandenberg Air Force Base Space Launch

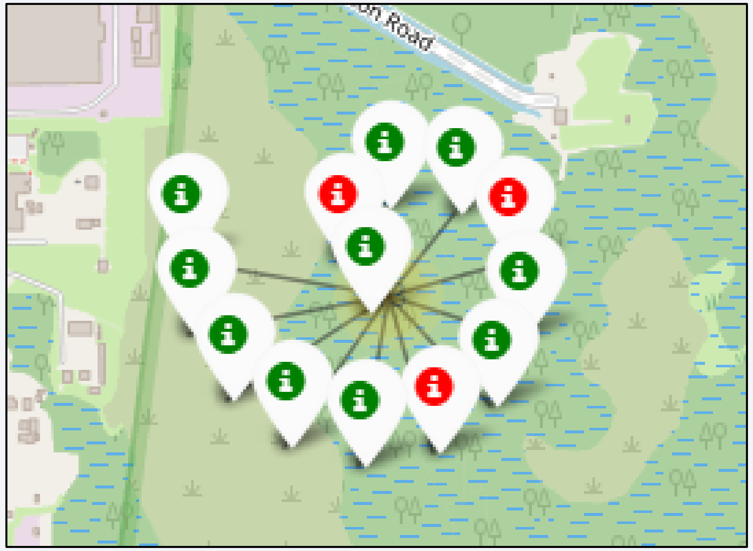


Cape Canaveral Space Launch
Kennedy Space Center Launch

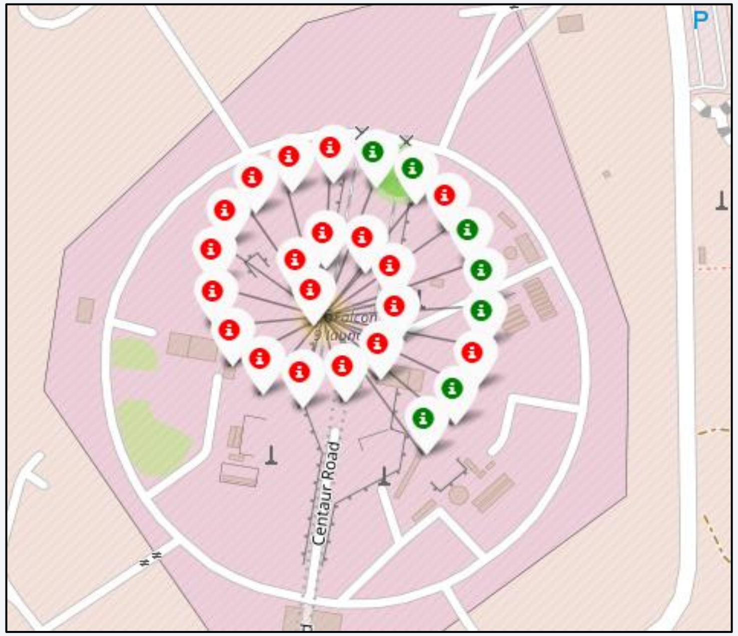
Folium Map – Color labelled launch outcomes



VAFB SLC – 4E



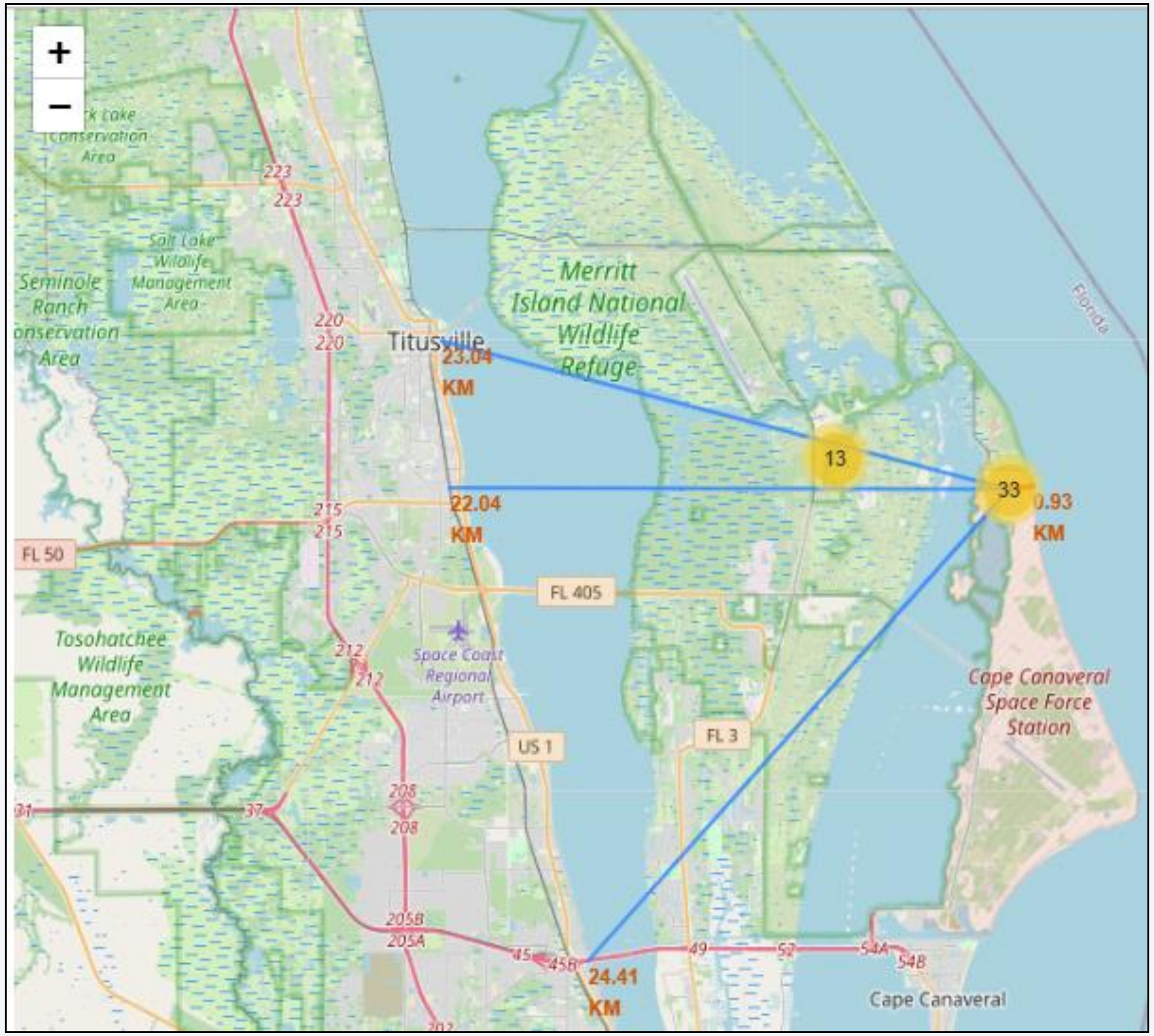
KSC LC-39 A



CCAFS SLC-40

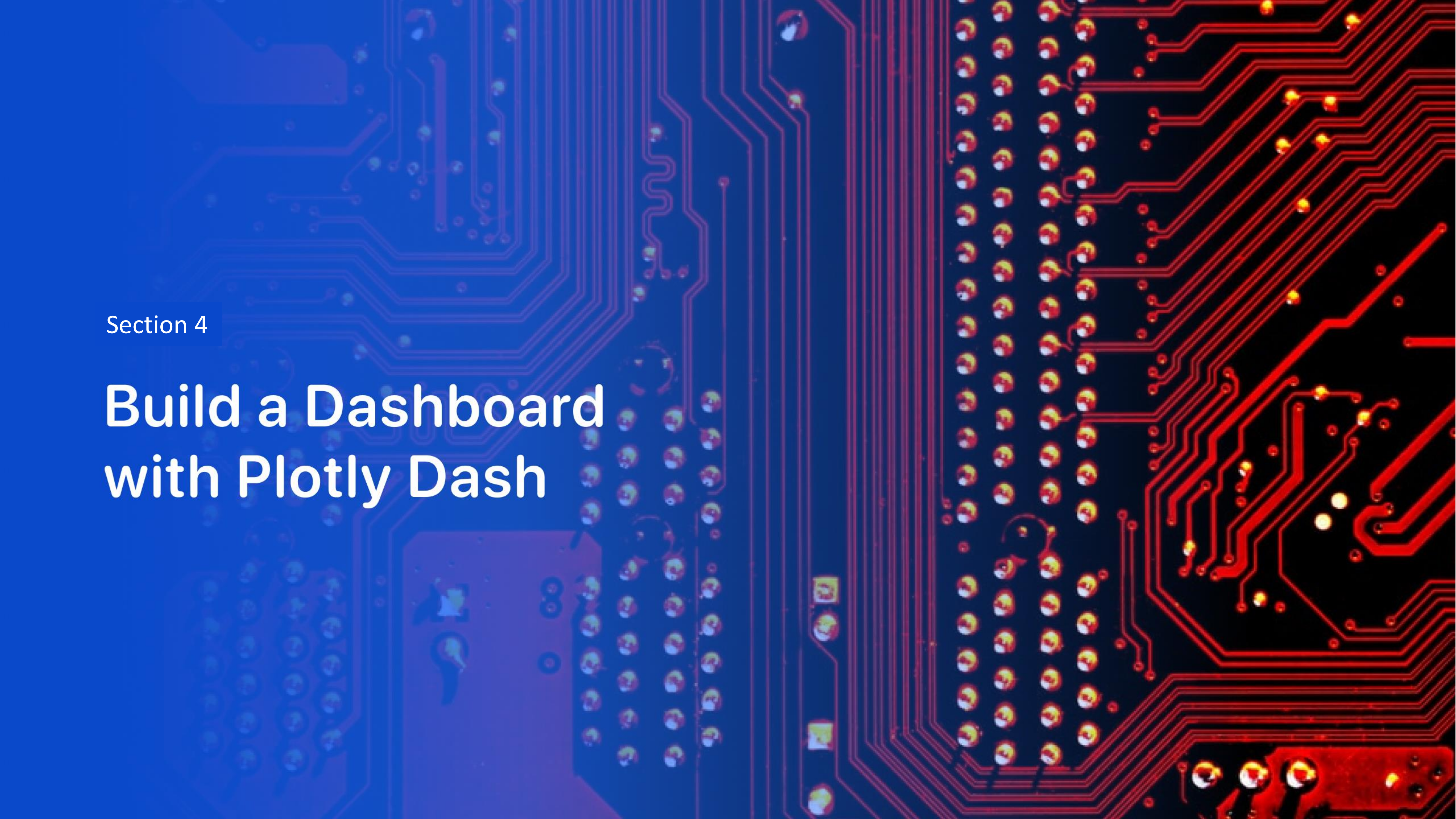
- Red icons refer to failed 1st stage landing
- Green icons refer to successful 1st stage landing

Folium Map – Distance between launch sites and proximities



Titusville
23.04 KM
13
33
0.93 KM
22.04 KM
24.41 KM
Merritt Island National Wildlife Refuge
Cape Canaveral Space Force Station
Space Coast Regional Airport
Cape Canaveral
FL 405
FL 3
US 1
FL 50
Seminole Ranch Conservation Area
Tosohatchee Wildlife Management Area
Salt Lake Wildlife Management Area

- Railway line that connects Titusville and Melbourne is 22.04 km from the launch site CCAFS SLC 40.
- Closest highway is Bennett Causeway and it's 24.41 km away from the launch site.
- Closest city to the launch site is Titusville which is 23.04 km away.
- Launch sites are far away from densely populated cities like Miami but very close to coastline (0.93 km) probably because of launch failures.

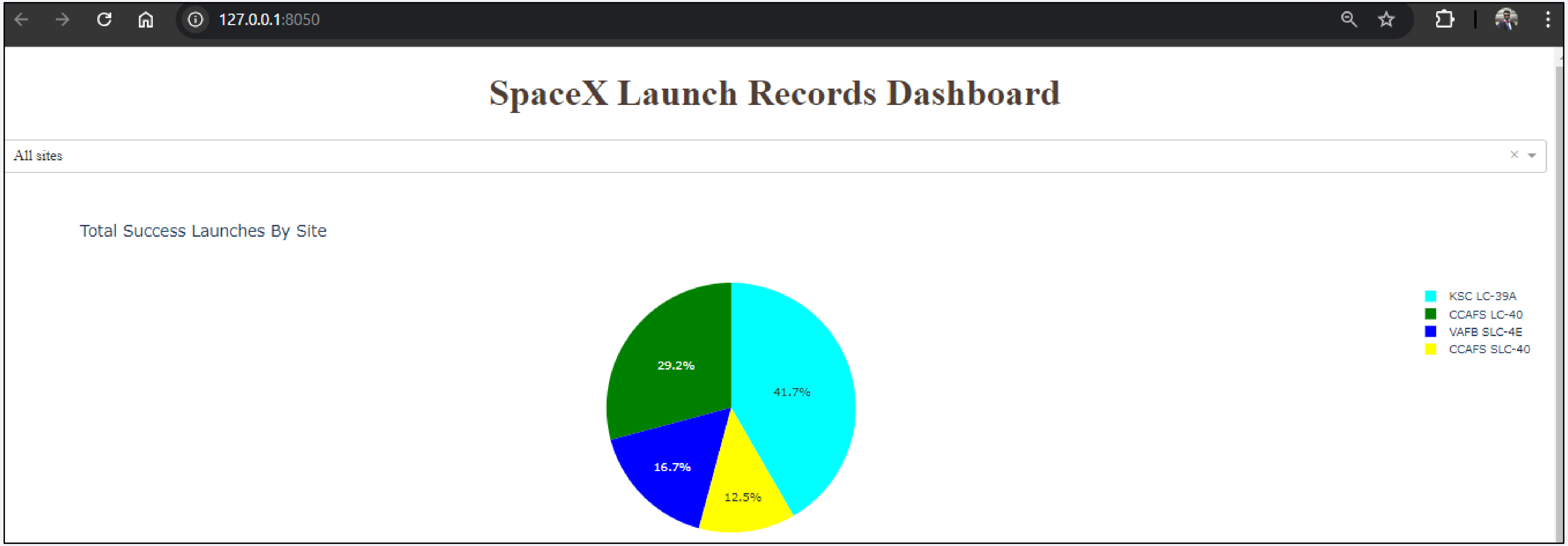


Section 4

Build a Dashboard with Plotly Dash

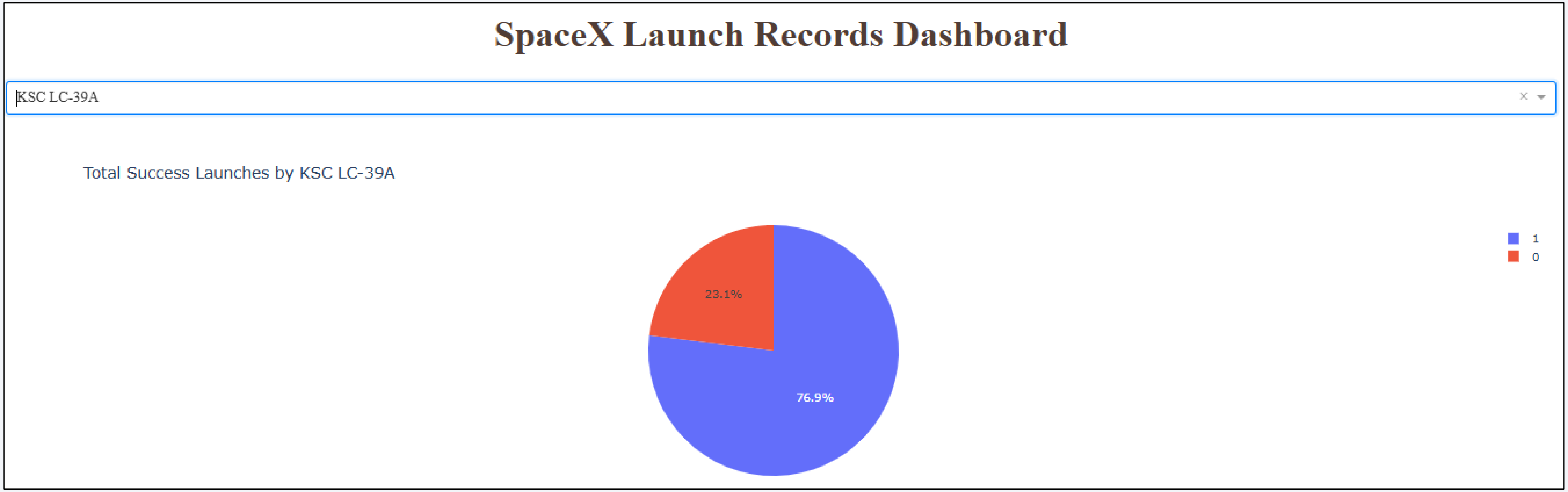
Dashboard – Pie chart for all sites

- When the drop down is not selected, then by default, launch records for all sites are selected.
- By default, success rate of all sites are displayed in a pie chart



Dashboard – Pie chart for site with highest launch success

- KSC LC-39A has the highest launch success ratio
- 10 successful 1st stage landing out of 13 launches resulting in 76.9%

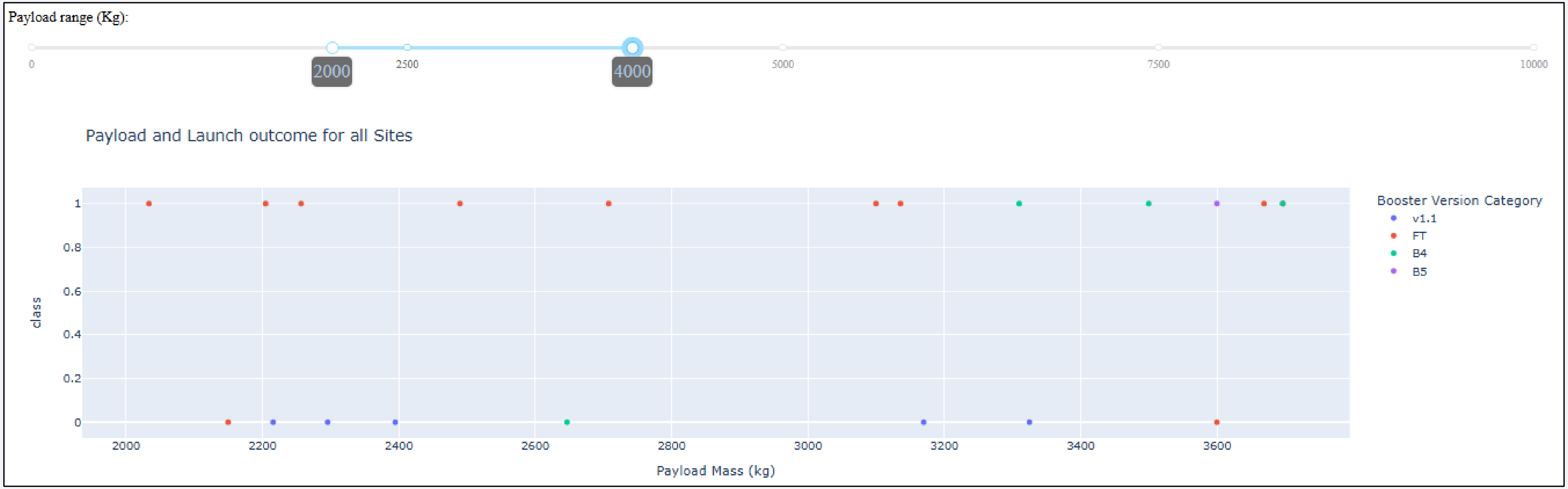


SpaceX Launch Records Dashboard

KSC LC-39A

Total Success Launches by KSC LC-39A

Dashboard – Scatter plot with Payload range slider



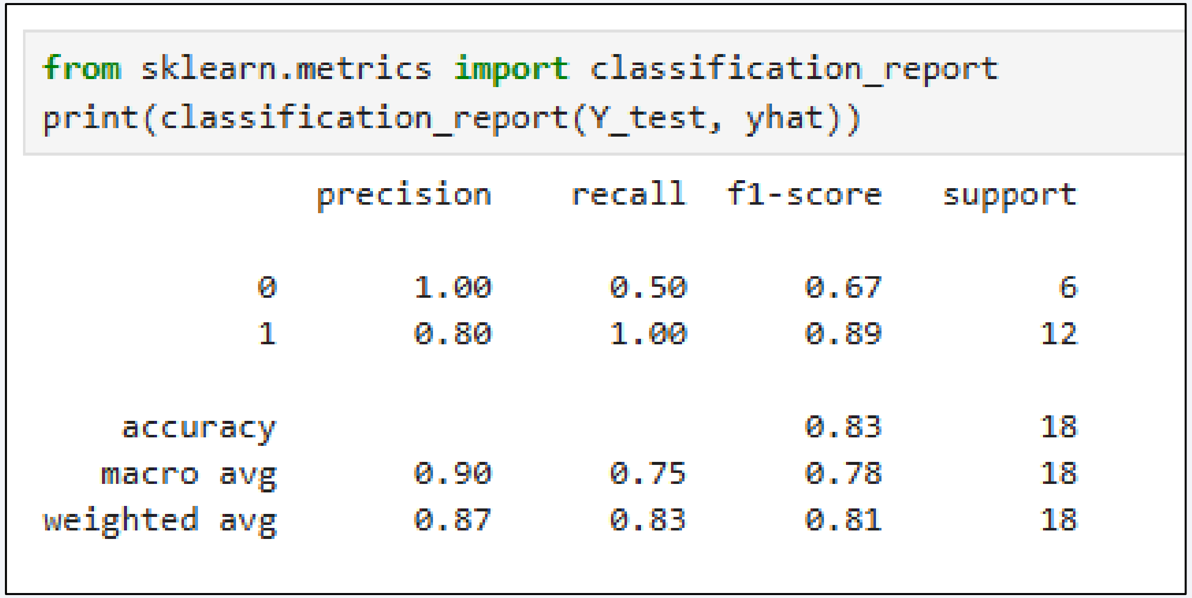
- Payload vs. Launch Outcome scatter plot for all sites, for payload between 2000 and 4000 kgs
- FT booster version has high success rate

Section 5

Predictive Analysis (Classification)

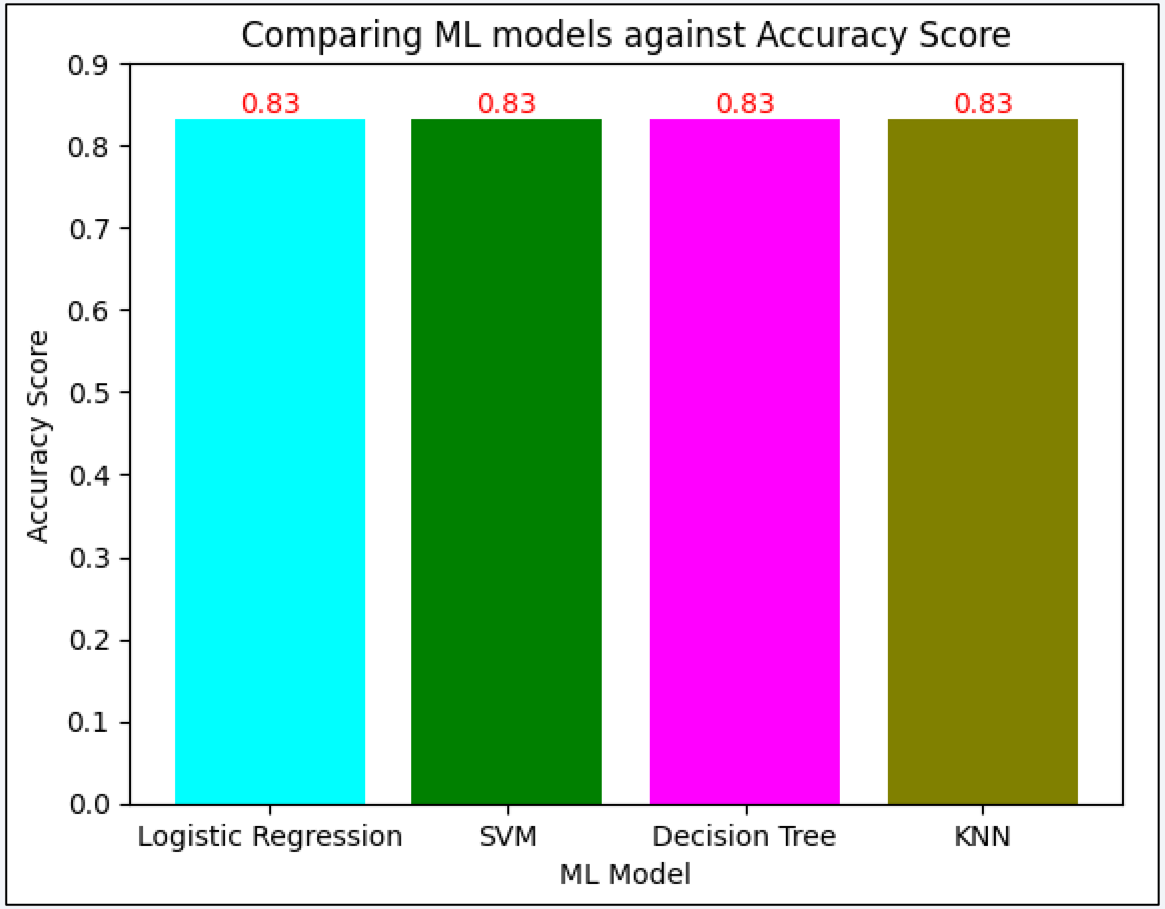
Classification Accuracy

- All ML classification models, have the same score of 0.83
- Since all the models have the same score, classification report summarizes precision, recall, f1-score and accuracy.



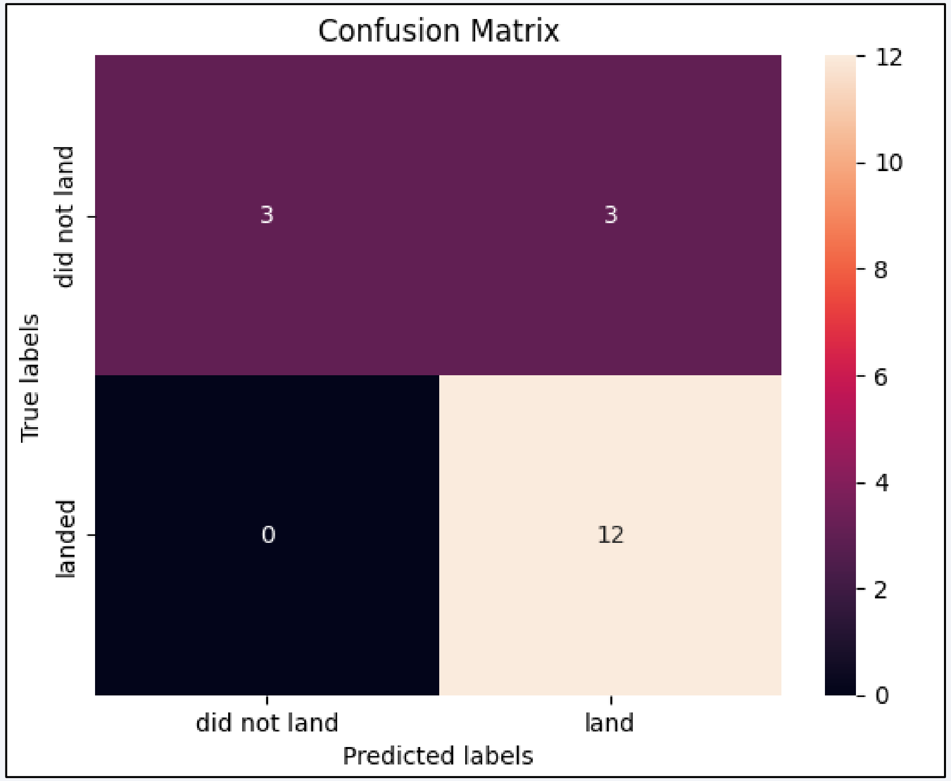
```
from sklearn.metrics import classification_report
print(classification_report(Y_test, yhat))
```

```
              precision    recall  f1-score   support

           0       1.00      0.50      0.67         6
           1       0.80      1.00      0.89        12

    accuracy                           0.83        18
   macro avg       0.90      0.75      0.78        18
weighted avg       0.87      0.83      0.81        18
```



Confusion Matrix

- Confusion matrix is the same for all ML models
- True Positive (TP) – 12 labels are truly landed and the model correctly classified 12 labels as landed
- True Negative (TN) – Model classified 3 labels as didn’t land which is correct.
- False Positive (FP) – Model classified 3 labels as landed but truly they didn’t land.
- False Negative (FN) – Model correctly classified 0 labels as didn’t land



$$Precision = \frac{TP}{TP + FP} = \frac{12}{15} = 0.8$$

$$F1 = \frac{2 * Precision \ * Recall}{Precision + Recall} = \frac{2 * 0.8 * 1}{0.8 + 1} = 0.889$$

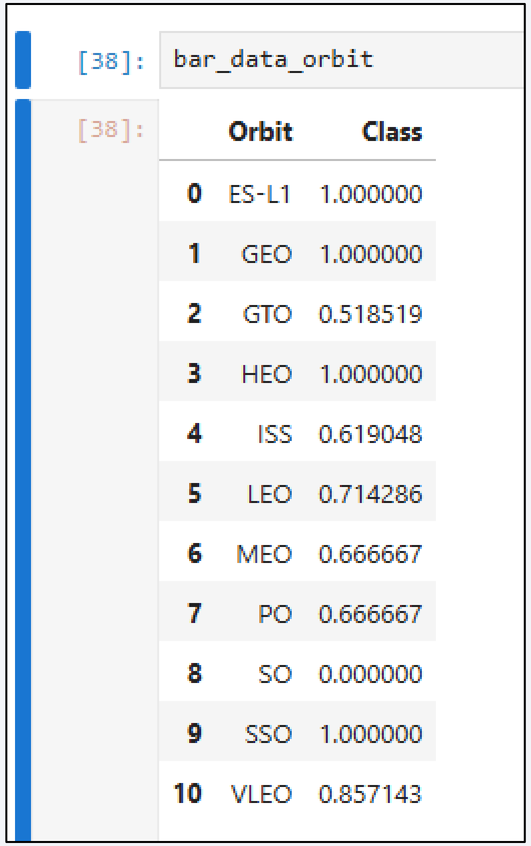
$$Recall = \frac{TP}{TP + FN} = \frac{12}{12} = 1$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{12 + 3}{12 + 3 + 3 + 0} = 0.83$$

Conclusions

- Launches from Kennedy Space Center Launch Site are relatively more successful
- Increasing number of 1st stage successful landing shows that feedback from previously failed outcomes are considered for improvements
- Launch sites are closely located to coastlines in case of failure launches to land in oceans away from civilian proximities.
- All ML models have the same score of 0.83 and the average f1 score is 78%
- The price the rocket launch can be predicted by estimating if the 1st stage would land successfully
- **Further work**
  - Domain experts should be consulted to confirm which independent variables are not needed in order to reduce data sparsity.
  - Create and fine tune a classification neural network model

Appendix

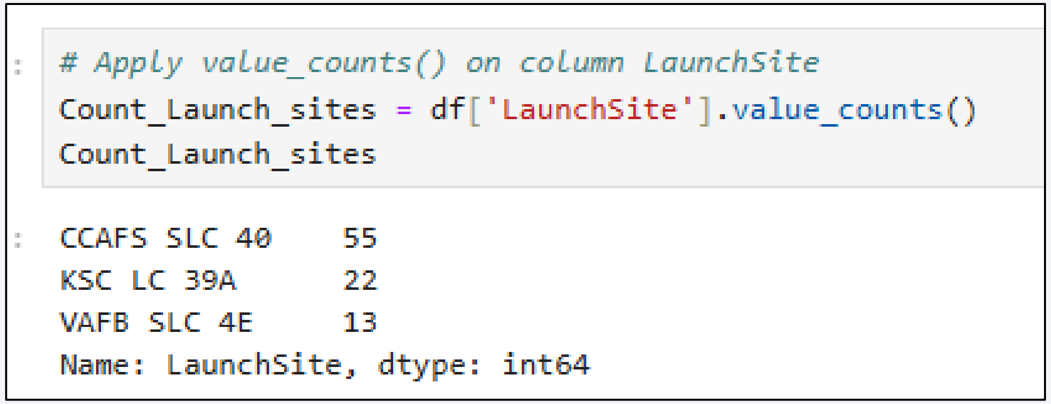


```
[38]: bar_data_orbit
```

[38]:

	Orbit	Class
0	ES-L1	1.000000
1	GEO	1.000000
2	GTO	0.518519
3	HEO	1.000000
4	ISS	0.619048
5	LEO	0.714286
6	MEO	0.666667
7	PO	0.666667
8	SO	0.000000
9	SSO	1.000000
10	VLEO	0.857143

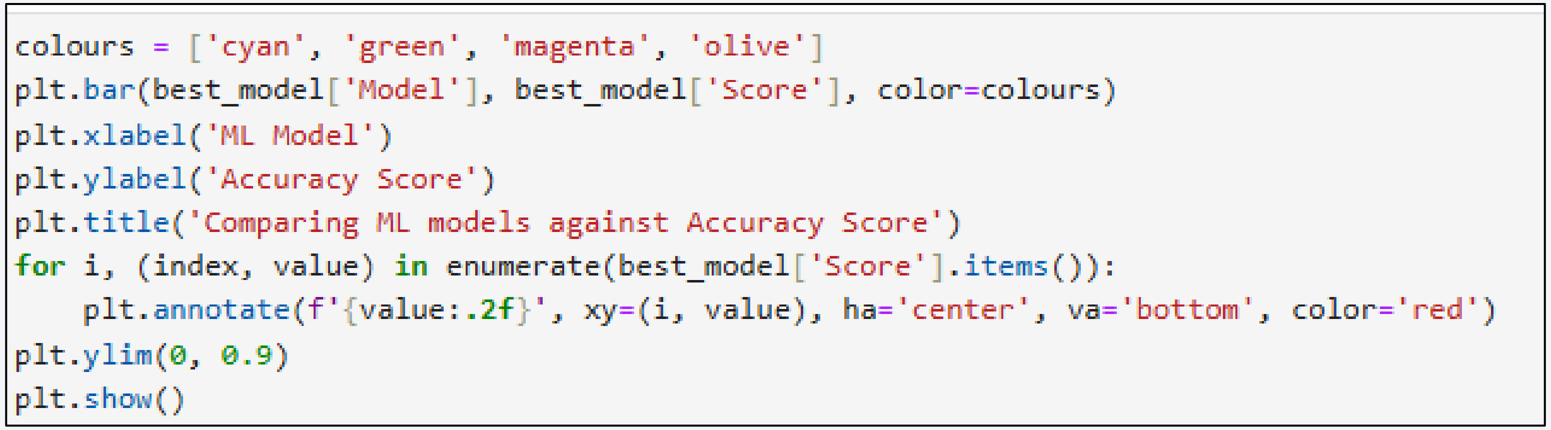
Success rate of orbit types



```
# Apply value_counts() on column LaunchSite
Count_Launch_sites = df['LaunchSite'].value_counts()
Count_Launch_sites
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

Count of launch sites



```
colours = ['cyan', 'green', 'magenta', 'olive']
plt.bar(best_model['Model'], best_model['Score'], color=colours)
plt.xlabel('ML Model')
plt.ylabel('Accuracy Score')
plt.title('Comparing ML models against Accuracy Score')
for i, (index, value) in enumerate(best_model['Score'].items()):
    plt.annotate(f'{value:.2f}', xy=(i, value), ha='center', va='bottom', color='red')
plt.ylim(0, 0.9)
plt.show()
```

Code snippet for bar chart comparing ML models

Thank you!